Documentation on the Open Smart Grid Platform
# Table of Contents

1. **Introduction**
2. **General platform architecture**
   1. Platform properties
   2. Architecture introduction
   3. Architecture functional layers
   4. Architecture Principles
   5. Platform components description
   6. Message flow examples
   7. Logical Authorisation Model
   8. Non-functional overview
      1. TimeBehavior
      2. Internationalization and localization
      3. Security
      4. Scalability
      5. Redundancy
      6. Performance
   9. **Technical Overview**
      1. Web Services Layer
      2. Domain Layer
      3. Core Layer
      4. Protocol Layer
      5. Technology Stack
   10. **Use cases**

   1. Installation Guide
      1. Installation
         1. **Vagrant**
         2. **Manual Setup**
      2. Platform Setup
      3. **Test the Platform**
         1. **Using SoapUi**
         2. **Using the Demo App**
   2. Configuration
      1. Add a device
      2. Users
      3. Add a new organisation
   3. Web Services
   4. Deployment
   5. **FAQ**

4. **Open Source Community**
   1. **Start contributing**
   2. Developers 101
   3. Contributing to the code
   4. Contributing to documentation
   5. Communication and Contact
   6. Governance
   7. Code of Conduct
   8. Foundation

5. **Domains**
   1. **Admin**
   2. **Smart lighting**
      1. Use cases
      2. Light Schedules
   3. **Tariff switching**
   4. Microgrids
   5. Distribution automation
   6. **SmartMetering**
      1. **bypass retry**
      2. priority
      3. scheduling
      4. AdHocManagement
         1. GetAssociationLnObjects
         2. GetGetAssociationLnObjectsResponse
         3. RetrieveConfigurationObjects
         4. GetRetrieveConfigurationObjectsResponse
         5. SpecificConfigurationObject
         6. SynchronizeTime
         7. GetSynchronizeTimeResponse
2. Response Messages
3. Use cases
7. Guidelines to add a new domain to GXF
6. Protocols
1. IEC61850
   1. SWDevice-010805
      1. SWDevice-010805.icd
      2. RegisterDevice
      3. GetConfiguration
      4. SetConfiguration
      5. SetEventNotifications
      6. EventNotification
      7. SetSchedule
      8. GetFirmwareVersion
      9. UpdateFirmware
     10. SetReboot
     11. StartSelfTest
     12. StopSelfTest
     13. SetLight
     14. SetTransition
     15. GetStatus
     16. UpdateDeviceSslCertification
2. FlexOVL_540_171101_2
   1. FlexOVL_540_171101_2_out.icd
   2. GetStatus
   3. EventNotification
2. DLMS / COSEM
1. DLMS device simulator
3. OSLP
   1. OSLP v0.5.1
      1. Protobuf Contract
   2. OSLP v0.6.1
      1. Protobuf Contract
      2. RegisterDevice
      3. ConfirmRegisterDevice
      4. GetConfiguration
      5. SetConfiguration
      6. SetEventNotifications
      7. EventNotification
      8. SetSchedule
      9. ResumeSchedule
     10. GetFirmwareVersion
     11. UpdateFirmware
     12. SetReboot
     13. StartSelfTest
     14. StopSelfTest
     15. SetLight
     16. SetTransition
     17. GetStatus
     18. UpdateDeviceSslCertification
     19. SetDeviceVerificationKey
     20. SwitchFirmware
     21. SwitchConfiguration
4. MQTT
7. Support
8. License
9. Glossary
Introduction

Introduction to Grid eXchange Fabric (GXF)

Grid eXchange Fabric was formerly known as the Open Smart Grid Platform (OSGP). GXF is an open, generic, scalable and independent 'Internet of Things' platform, which enables various connected smart objects in the public space to be easily controlled and monitored. Our platform allows the use of any (web)application and with any IP communication infrastructure.

Our goal is to stimulate the development of smart and sustainable solutions. Smart devices and smart apps play a central role in the development of smart grids and smart societies. The open smart grid platform software enables you to connect to thousands of devices, control them, and monitor their performance. This is done in an open and secure way, so you can use it for your own applications and devices, thereby reducing the time to market and decreasing development costs.

The names Open Smart Grid Platform and Grid eXchange Fabric will both be found in this document and in the code. In the future we will replace all references to the Open Smart Grid Platform and OSGP by Grid eXchange Fabric and GXF. During this transition period please read Grid eXchange Fabric if you see Open Smart Grid Platform.

Using the platform

The Grid eXchange Fabric is used in the following way:

- A user or operator uses one or more (web) applications to monitor and/or control devices
- The applications connect to the open smart grid platform via several web services which are divided into functional domains, i.e. Public Lighting, Smart Metering, Power Quality, etc. Third party developers can use the web services for the development or integration of new applications
- The platform handles all these application requests in a secure way and uses various functions and services to do so (e.g. authorization, authentication, device management, logging)
- For the 'translation' and communication of user/operator commands to the various smart devices, the platform uses multiple (open) protocols
- The platform supports various IP based data telecommunication technologies and protocols to communicate with the devices

More technical and user information about Grid eXchange Fabric can be found in this document. More generic/product information about Grid eXchange Fabric can be found on the Grid eXchange Fabric website.

Example use case for Grid eXchange Fabric

Use cases of Grid eXchange Fabric are only limited by your imagination. Here are some examples:

- Ad-hoc and scheduled Switching of Public Lighting
- Electrical Transportation
- Smart Metering
- Traffic Control (LED/matrix signs, traffic lights)
- Flexible load management (solar energy and wind energy)
- Power Quality monitoring

Getting started

- Visit the userguide section to try Grid eXchange Fabric on your local machine
• The Architecture section provides information on platform architecture
• Check out the domain section if you want to know about the existing domains
• Check out the protocol section to find out more on the existing supported protocols
• Read the open source section how to contribute!
General platform architecture

CHAPTER 1 Technical overview of the Open Smart Grid Platform

This chapter contains the general architecture and properties of the Open Smart Grid Platform. Domain and protocol specific information can be found in the domain and protocol chapters. This chapter is written for (potential) users, architects and developers.
Platform properties

Properties of the Open Smart Grid Platform

The Open Smart Grid Platform is designed for message based communication.

- Acts as a connecting link between (web)applications and smart devices
- The open Source approach prevents vendor lock-in
- State of the art security
- Fully scalable, dynamically scaling up and down as more devices and applications are added.
- Freedom of choice in the desired IP communication infrastructure, e.g. CDMA and GPRS.
- Stimulates open innovation by using open standards and open source technology
- Multiple devices and communication protocols are supported
- Independent of (cloud) hosting infrastructure
- By de-linking the chain and the use of open standards and the open source license, anyone can build his or her applications on top of the open smart grid platform.
- The open smart grid platform is optimized to provide reliable and efficient delivery of command and control information for e.g. smart meters, direct load control modules, solar panels, gateways and other applications.
- The open smart grid platform simplifies the implementation of smart devices resulting in a shorter time-to-market by having built-in device management features
- The platform supports various IP data communication infrastructures to communicate with the devices (internet, lan, GPRS, CDMA, UMTS, etc.).
- The open smart grid platform also supports authentication and encryption for all data exchanges to protect the integrity and privacy of data as required in e.g. the smart grid.
- The open smart grid platform supports multiple protocols
- Easy application integration
- Supports active-active setup over multiple data centers
- Adding servers can be done in runtime

Please note: the Open Smart Grid Platform is not built for streaming data such as video, audio or a stream of high frequency measurement data.

Unique features of the Open Smart Grid Platform

The Open Smart Grid Platform is unique due to its multi-dimensional, generic and open design. Because of a true separation of layers and the use of open standards, other suppliers and/or third parties are able to develop and market innovative solutions.

1. The platform is multi-dimensional. This means that several customer use cases (with separate business models) are able to use the various device functions. One single application could use the same function of different devices.
2. The generic design ensures that the platform can be used in a flexible way for several functions and applications (e.g. public lighting services and smart meter services).
3. The platform is aimed at the ‘common parts’ of the technology chain; suppliers or vendors (of both applications and devices) have no competitive advantage in delivering these kind of services.
4. The platform layers are truly separated by open standards and the platform is made available as open source software.
5. The platform does not store any application data (the platform is thus stateless). No messages/commands will ever get lost. This enables third party vendors and developers to deliver innovative applications which are competitive in both rich functionalities and the generated data.
Architecture introduction

Basic Architecture

The basic architecture
Basic Overview

Layered architecture
The Open Smart Grid Platform environment consists of five layers:

1. Web services layer
2. Domain logic layer
3. Open Smart Grid Platform Core layer
4. Protocol layer
5. Device layer

**Web services layer**

In this layer the web services are exposed to the outside world. Applications can connect to the web services to implement the required functionality of the open smart grid platform. The web services are divided into functional domains, i.e. Public Lighting, Smart Metering, Power Quality, etc. Additional functional domains can be created.

**Domain logic layer**

Every functional domain has a separate set of web services and a corresponding domain logic block. In the domain logic block the business logic of that functional domain can be found. This is where a functional command will be translated into a generic intermediate format. For example, in the case of public lighting the command "Turn light on" will be translated into a command like "set switch(1) in closed position". In this layer it could also be decided that one functional command results in multiple commands to a device. The domain logic is closely related to the web services layers and can be added as well.

**Open Smart Grid Platform core layer**

In the core of the Open Smart Grid Platform the following generic functions are found:

- Device management
- Time synchronization
- Firmware management
- Workflow engine
- Device installation services
- Scheduler
- Device status monitoring
- Routing of device commands to appropriate device protocol

**Protocol layer**

The different protocol adapters are found in this layer. Here the generic intermediate format of a command for a specific device will be translated into the protocol message the device understands. This message will be sent to the device. A retry mechanism has been implemented to prevent communication failure in the case that the receiving end is temporarily unavailable. The listeners for messages initiated by a device are implemented here. Examples are the DLMS/COSEM protocol adapter for smart meters.

**Devices**

Any device in the public space with an Internet connection may be connected to the platform. The platform is independent of the device used, therefore this part of the set-up is not part of the platform.
Architecture functional layers

Functional view

The Functional view shows an overview of the most important functions of the system. The two images below show the starting architecture and functional reference architecture respectively.

Starting architecture

The Functional view shows an overview of the most important functions of the system. The two images below show the starting architecture and functional reference architecture respectively.
1. Web applications
2. Open Smart Grid Platform
3. Web services
4. Basic functions
5. Database
6. Communication infrastructure (CDMA/GPRS/Ethernet)
7. IP infrastructure
8. Open Street Light Protocol (OSLP)
9. Public Street Lighting Device (PSLD) or Sub Station Lighting Device (SSLD)

Functional Reference

This model partitions the system in seven functional clusters (vertically) which are shown on the system layers (horizontally). The circled numbers refer to image 1.

Image, functional reference architecture

Vertical clusters:
- Device installation
- Device management
- Firmware management
- Configuration management
- Schedule management
- Ad-hoc control and status
- Monitoring

Horizontal System layers:
- Web applications
- HTTPS/SOAP communication
- Platform
- Open protocols
- Smart devices
Architecture Principles

This chapter gives an overview of the principles used defining and implementing the architecture. The following principles were applied:

- Layering
- **Domain driven design**
- Dependency inversion principle
- Behavior driven development

**Layering**

The use of layers improves the separation of responsibilities. Each application contains the following layers:

- Presentation layer: responsible for providing information to users (persons and/or systems) and the handling of user requests
- Application layer: responsible for executing system tasks including authorisation control
- Domain layer: responsible for the representation of the problem domain.
- Infrastructure layer: responsible for technical matters supporting other layers. For instance persistence, messaging, etc.

![Image, Layers:](image_url)

1. Audit logger
2. Web Services
3. Functions
4. Queue
5. Workflow engine
6. Protocol framework
7. Protocol implementations
8. Workflow engine
9. Queue
10. Communication
**Domain driven design** (DDD)

Domain-driven design focuses on the problem domain. DDD's starting point is creating an optimal model for a specific problem domain by having a common language and constructive collaboration between technical and domain experts.

DDD uses the following building blocks:

- **Entity**: An object not identified by its attributes but by its own identity.
- **Value Object**: An object with attributes but has no own identity.
- **A collection of objects surrounding a specific root entity (or aggregate root)**. To ensure consistency objects in the aggregate can only be addressed through the aggregate root.
- **Service**: Contains instructions not related to a specific object.
- **Repository**: Serves as a collection for fetching and saving objects. Creates an abstraction for actual persistent implementations.
- **Factory**: Contains methods to create domain objects.

**Dependency inversion principle**

The dependency inversion principle promotes an independent connection by inverting dependency relations. This ensures that the domain model can be very 'clean' without knowledge of the underlying infrastructure (POJO classes). The Spring framework is used to implement the Dependency Inversion principle.

**Behavior driven development** (BDD)

Behavior driven development is a way of programming that first describes behavior in user stories and then implements this in code. The user stories contain scenarios with acceptance criteria that can be automated. This creates a complete test suite for the whole system.

For the application of BDD the following frameworks are used:

- Cucumber and Gherkin, automated acceptance testing, based on scenarios from stories.
Platform components description

Description of the individual platform components

Application Layering

The use of layers improves the separation of responsibilities. Each application contains the following layers:

- **Presentation layer**: responsible for providing information to users (persons and/or systems) and the handling of user requests
- **Application layer**: responsible for executing system tasks including authorization control
- **Domain layer**: responsible for the representation of the problem domain.
- **Infrastructure layer**: responsible for technical matters supporting other layers. For instance persistence, messaging, etc

Layers:

![Diagram of layers](image-url)

Authentication and authorization

The web server is configured with a SSL certificate to encrypt the incoming and outgoing communication. The SOAP Web service (Spring Framework web service) uses a Java Keystore and a certificate for each organization. Only organizations that are known within the platform are authorized to use the web service.

Application integration layer

For the several functional domains separate SOAP Web services are offered. This separation offers authorization per functional domain. Each of the web service components send a queue message to the corresponding domain component.

WSDL A separate WSDL is implemented for each functional cluster. All SOAP operations have a request object parameter and return a response object. For Synchronized Web Services the result is immediately included in the response. For asynchronous web services the response contains a correlation ID. This Correlation ID is to be used by the requester to receive the actual result from the platform. The following diagram is an example of such an asynchronous request.
Furthermore each SOAP message has a header which contains the user’s organisation ID. This table displays an overview of the WSDL’s including operations and fields in the request and response objects.

**SOAP vs. REST**

SOAP is chosen in the open smart grid platform web services over REST for the following reasons:

- REST is resources/data oriented (put, get, delete) while the open smart grid platform is function/method oriented
- SOAP has the advantage of having a contract (WSDL)
- SOAP has extensive security features that are being used in the open smart grid platform to meet the high security demands/requirements requested by e.g. the energy utilities
- Energy companies are generally not progressive in terms of technology. SOAP is acceptable for energy companies and REST is sometimes seen as new and insecure.

The benefits of REST (e.g. speed / less overhead) does not outweigh the benefits of SOAP. More general information on this topic can be found online.

**Domain logic layer**

For each functional domain business logic is implemented using a separate domain component. Common functionality like authorization should be abstracted to a shared component. Domain components receive queue messages from web service components and send queue messages to the open smart grid platform core component.

More information on the specific domains can be found in the domain chapter.

**Open Smart Grid Platform Core Services**

The open smart grid platform core component receives queue messages from domain components. These messages from domain components are forwarded to a protocol adapter project. The open smart grid platform core component also offers logic for a protocol adapter project to send the response of a smart device back to a domain project. The Core component routes messages from domain adapter components to protocol adapter components and vice versa. The core layer also contains a workflow engine.

The internal database model in the core layer:
ERD’s made with Valetina Studio

Overview of platform data model:

Platform components description
Data model explanation:

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>devices</td>
<td>Devices table</td>
</tr>
<tr>
<td>device_authorisation</td>
<td>Authorisation table, function group column concerns the device functions (AD_HOC, INSTALLATION, etc)</td>
</tr>
<tr>
<td>organization</td>
<td>Organization table, function group column concerns the platform functions (ADMIN of USER)</td>
</tr>
<tr>
<td>event</td>
<td>Events table</td>
</tr>
<tr>
<td>oslp_log_item</td>
<td>Table for logging of OSLP messages.</td>
</tr>
<tr>
<td>webservice monitor log item</td>
<td>Audit record for tracking webservice activity.</td>
</tr>
</tbody>
</table>

The platform will store as little data as possible. Generic (and domain specific) devices attributes are stored in core DB.

**Protocol Layer**

The open smart grid platform supports multiple protocols.

- **OSLP** (Open Street Light Protocol)
- **DLMS/COSEM**
- **IEC61850**

The protocols can use one of the security layers:

- **TLS** (Transport Layer Security encryption)
- **SSL** (Secure Sockets Layer encryption)

Other protocols can be easily added to the platform. If possible, we prefer protocols based on open standards. A comprehensive list of protocols that are currently supported can be found in the **protocols chapter**.

Protocol specific device attributes are stored in the protocol adapter DB

**Queues**

Open smart grid platform components connect to each other through message queues.
Transactions on messages to and from the queues
Messages are persisted on the queues
Queues are clustered for reliability and speed
By using queues, the open smart grid platform can be stateless

Smart devices
The open smart grid platform can connect to any device that supports one of the supported protocols. Smart devices can receive messages from or send messages to protocol adapter components. In case of SSLD’s this is done using TCP/IP over mobile internet connections (e.g. GPRS, CDMA, etc.). The communication is encrypted using public key cryptography.
Message flow examples

This are some examples how a message flows in the Open Smart Grid Platform.

Message Flow: Request/Acknowledge/Poll

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>WS adapter receives client soap request with organization certificate and organization id in soap header</td>
</tr>
<tr>
<td>2</td>
<td>WS adapter authenticates organization, checks authorizations and sends request message to domain adapter (via queue)</td>
</tr>
<tr>
<td>3</td>
<td>WS adapter returns soap acknowledgement with correlation id</td>
</tr>
<tr>
<td>4</td>
<td>Domain adapter sends request message to core (via queue)</td>
</tr>
<tr>
<td>5</td>
<td>Core determines protocol for device and sends request message to protocol adapter (via queue)</td>
</tr>
<tr>
<td>6</td>
<td>Protocol adapter translates domain request message, sends request to device and receives response from device</td>
</tr>
<tr>
<td>7</td>
<td>Protocol adapter sends response message to core (via queue)</td>
</tr>
<tr>
<td>8</td>
<td>Core forwards response message to domain adapter (via queue)</td>
</tr>
<tr>
<td>9</td>
<td>Domain adapter forwards response message to response queue</td>
</tr>
<tr>
<td>10</td>
<td>Client app polls for response using correlation id (with organization certificate and organization id in soap header)</td>
</tr>
<tr>
<td>11</td>
<td>WS adapter retrieves response message from response queue</td>
</tr>
<tr>
<td>12</td>
<td>WS adapter sends soap response to client app</td>
</tr>
</tbody>
</table>

Message Flow: Request/Acknowledge/Notify
### Step Description

1-8  Same as request/acknowledge/poll message flow  
9   Domain adapter forwards response message to WS adapter  
10  WS adapter stores response in DB  
11  WS adapter sends soap notification with correlation id to the client app*  
12  Client app sends soap request with correlation id to retrieve the response  
13  WS adapter retrieves (and deletes) response from DB  
14  WS adapter sends soap response to the client app  

*In case the response is not timely retrieved by the client app, OSGP will resend the notification with correlation id to the client app. The amount of retries is configurable.
Logical Authorisation Model

Logical Authorisation Data Model

Authentication of open smart grid platform

The Open Smart Grid Platform contains an extensive authorization model, which enables a device owner to give certain rights on certain devices to other organizations. Every organization will only see devices they have rights to.

This model displays the most important entities of the open smart grid platform system and their mutual relationships.

Image of Logical Authorisation Data Model

The logic of the model above:

At the top of the image is the entity "authorisation". This represents the permissions of an organization on a certain device. In general an organisation will have a lot of permissions, at least one for each device it needs to manage.

The functions an organisation can execute on a device are determined by the function group the authorisation refers to. Function groups are collections of functions and are predefined in the software. The following function groups have been predefined:

- Owner-group (this contains all functions)
- Ad hoc-group (Functions for ad hoc switching of lighting)
- Management-group (Platform functions)
- Installation-group (Functions to install devices)
- Firmware-group (Functions for updating firmware)
- Schedule-group (Functions to create lighting schedules)
• Tariff scheme-group (Functions to create tariff groups)
• Configuration-group (Functions to configure devices)
• Monitoring-group (Functions to monitor devices)

This structure provides maximum flexibility when assigning rights to devices. Devices always belong to an Owner. An owner is an organisation, but not every organisation is an owner. A device can have more than one owner. The entity "Event", at the bottom of the image, is the execution of a function by an organisation on a device.

Details like device-type, device-status, etc. have been omitted from this model.

One security requirement is that each event must be traced back to a 'natural' person, also known as an audit trail. Although the open smart grid platform does not register individual users we can meet this requirement by registering a data-item with each event. This enables the user organisation to investigate which events belongs to which 'natural' person. This data-item can for example be an user-ID provided by the user organisation which doesn't have to be unique in the open smart grid platform.

Table describing the entities in the logical data-model

<table>
<thead>
<tr>
<th>Entity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>authorization</td>
<td>Authorization – Permissions of an organisation to execute a certain function (member of a role) on a certain device</td>
</tr>
<tr>
<td>device</td>
<td>Electronics present in a &quot;container&quot; (for example a lamp post), connecting to the open smart grid platform and (in case of a lamp post) controls the lights. One device has one owner.</td>
</tr>
<tr>
<td>audit trail</td>
<td>The actions of an organization on a device. A combination of [time, organization, function and device].</td>
</tr>
<tr>
<td>function</td>
<td>An end-to-end operation. For example &quot;set schedule&quot;. A function belongs to multiple function groups.</td>
</tr>
<tr>
<td>function group</td>
<td>Usually this will be larger groups than the “function clusters” in this document. All functions available to end users could be in a single group for example.</td>
</tr>
<tr>
<td>organization</td>
<td>A party playing a role in the management and control of the devices, for example municipalities.</td>
</tr>
<tr>
<td>owner</td>
<td>An organization role. Each device has an owner.</td>
</tr>
<tr>
<td>event</td>
<td>An event reported by the device (which is not the action of a natural person) for example an error or security problem.</td>
</tr>
</tbody>
</table>

An organization can get rights to one or more function groups, and thus all functions in that function group will be available to this organization.

To ensure that devices can only receive instructions from a 'genuine' open smart grid platform it must be possible to authenticate the open smart grid platform. This is implemented through a standard technology based on asymmetric encryption (if supported by the Device). The open smart grid platform will receive a unique key to enable the devices to tell if the messages come from a 'genuine' open smart grid platform. Both OSLP and DLMS device types use this kind of encryption. To prevent replay-attacks each message will get an index number (this is standard practice as well).

Authentication of devices

To ensure that the open smart grid platform can distinguish between 'genuine' devices and 'illegal' devices, all devices are supplied with a manufacturer key. Each device has a unique key. Because of the asymmetrical encryption the platform contains the public part of each key. In this way devices can be identified by their unique key and their unique hardware ID.

The device-ID will be encrypted in each message sent from the device to the platform.

All communication between the open smart grid platform and the devices will be signed with these keys to ensure (1) the source is legitimate and (2) to ensure the integrity of the message. It is not necessary to encrypt the whole message because confidentiality is not important. This results in a less computationally intensive process.

When a key is stolen (by hacking a device) this will not affect the integrity of the other devices. Each device has an unique key after all and only the hacked device has to be excluded from communication in the platform.

The security is independent from the carrier (GPRS, CDMA, Ethernet, etc.). The open smart grid platform supports symmetric and asymmetric encryption (depends on device and protocol).

For OSLP devices, the firmware will be used to distribute keys to devices. In this way we can use the existing secure firmware update mechanism for updating keys and certificates. DLMS devices use a mechanism to switch keys that is not dependent on firmware updates.

Additional security may be provided by using TLS communication.

Authorisation of organisations

Authorisation for use of the platform functionalities is handled by function groups. Function groups are defined for both platform functionality and device functionality. Each function group has one or more functions. Access to device functions can be set per device. The tables below show an overview of all function-groups and device-functions and platform-groups and platform-functions respectively.
<table>
<thead>
<tr>
<th>Functions</th>
<th>OWNER</th>
<th>INSTALLATION</th>
<th>AD_HOC</th>
<th>MANAGEMENT</th>
<th>FIRMWARE</th>
<th>SCHEDULING</th>
<th>TARIFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET_DEVICE_AUTHORISATION</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>SET_DEVICE_AUTHORISATION</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>START_SELF_TEST</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STOP_SELF_TEST</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SET_LIGHT</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GET_STATUS</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RESUME_SCHEDULE</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SET_REBOOT</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SET_TRANSITION</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SET_EVENT_NOTIFICATIONS</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GET_EVENT_NOTIFICATIONS</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REMOVE_DEVICE</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UPDATE_FIRMWARE</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>GET_FIRMWARE_VERSION</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SET_SCHEDULE</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SET_TARIFF_SCHEDULE</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SET_CONFIGURATION</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GET_CONFIGURATION</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GET_ACTUAL_POWER_USAGE</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GET_POWER_USAGE_HISTORY</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Groups</th>
<th>ADMIN</th>
<th>USER</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATE_ORGANISATION</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>GET_ORGANISATIONS</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>GET_DEVICE_NO_OWNER</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>GET_MESSAGES</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>FIND_DEVICES</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SET_OWNER</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
Non-functional overview

Non-functional view

The non-functional view is an overview of the most significant non-functional demands.

The identified non-functional demands are:

- Time Behavior
- Extensibility
- Internationalisation and localisation
- Security
- Scalability
**TimeBehavior**

**Time Behavior**

Time behavior is mainly important in the Flexovl application when a lot of devices have to be addressed in a short period of time over a wireless network. Both latency and limited bandwidth have to be taken into consideration while demanding the coordinated on and off switching of the lighting, since we want to avoid the Christmas tree effect.

- Time synchronization: devices periodically register with the platform and receive a time.
- Protocol: because of the limited bandwidth an efficient protocol "protobuf" was selected.

**Points of interest:**

- Light metering messages
- When the SSLD's are disabled the PSLDs cannot be addressed

Because of these points of interest we use message queueing combined with a retry mechanism of delayed delivery.

The platform and devices use UTC time. The OSLP protocol between platform and devices uses UTC time as well.
Internationalization and localization

The platform and devices use UTC time. The OSLP protocol between platform and devices uses UTC time as well.
Security

Cloud security

- DDOS protection
- IPSEC VPN connections
- IP whitelisting

Most cloud environments support these features.

Operating System

- Hardened operating systems (according to Center of Internet Security)

Platform security

- Communication over TLS
- Firewalls between all servers and layers
- Certificates from a recognized Certificate Authority (CA)
- Audit trail on all actions throughout the platform
- Role based authorizations on specific functions of devices
- Access control
- Unique device identification

For every major release there will be a mandated security test initiated by Alliander.

In cooperation with the European Network of Cyber Security (ENCS) state of the art security measures were implemented.

- Security per device
- Security per application
- Security Certificates per Organisation and per device
- All communication is encrypted
Security measures:

1. Firewall in defined zone
2. Operating System Hardening
3. DDOS protection
4. Replay attack prevention
5. Private encryption key per device
6. Certificates from a Certificate Authority
7. Encryption via Elliptic Curve DSA
8. IPSEC VPN for CDMA and GPRS
9. Unique device identification
10. Unique CDMA modem number
11. Role based authorizations on functions and devices

Encryption

An analysis of safety aspects has led to the decision that the safety of the whole system will be realized by proven technology based on asymmetrical coding (also known as public-key encryption).

Authentication of web applications

Two-way SSL will be used between web applications and the Open Smart Grid Platform to verify the identities for both client and server. User organisations are responsible for the administration of the identity of and access to their web applications. The web applications feature a login page. After successful login the user is linked to an organisation. Passwords will be stored with encryption. The organisation ID will be sent in each message to the Open Smart Grid Platform and will be verified by the SSL certificate.

Algorithms

Only public encryption Algorithms will be used. Due to performance limitations (of the devices) and recommendations from
The European Network for Cyber Security (ENCS) Elliptic Curve DSA with 256-bit-keys was selected. This improves the
security and efficiency over the 1024 bit RSA algorithm. Messages can be smaller and less processor capacity is needed. The
key length of Elliptic Curve DSA is similar to the 3072 bit key length of RSA.

Note: Even though the open smart grid platform uses ECDSA to secure the OSLP, other encryptions may be used as well.
The RSA Algorithm is still supported if preferred. This is a flexible configuration option.

Private APN

A private APN is used for linking to mobile data communication infrastructures.

Logging

- Every action to and from devices is logged in the audit trail
- Messages from unknown devices will be denied (and logged)
Scalability

The Open Smart Grid Platform is designed and built for scalability and reliability:

- Messages will never get lost. In the worst case scenario, a message will be sent to the dead letter queue.
- Any layer of the platform can be independently scaled up- and down.
- Adding servers can be done runtime.
- It can run in an active-active setup over multiple servers and data centers. In our cloud hosted setup even over sets of data centers in different countries.
Applications

Availability zone 1

Webservices

Domain logic layer

OSGP Core

Protocol layer

Availability zone 2

Webservices

Domain logic layer

OSGP Core

Protocol layer

Load Balance

Scale out

Devices
Redundancy

This chapter describes the possibilities of a redundant set up of the Open Smart Grid Platform. The Platform is designed to run in a High Available (or HA) environment, and to prevent data loss due to unexpected failures. Each component of the Platform is designed to be stateless. Components communicate with each other using message queues, which are processed in an asynchronous way.

Active-active

In an active-active setup, multiple instances of each component (e.g. Web Services, Core, Protocol Adapter) process the data at the same time. Traffic is equally distributed across the instances. In case of a defect in one instance the traffic is automatically processed by the remaining instance(s). The Open Smart Grid Platform is designed to run in such a set up, and thus preventing down time in case of a failing server. Each component of the Platform can run in an independent, redundant and scalable way.

Database

The Open Smart Grid Platform uses a PostgreSQL database. PostgreSQL supports multiple database servers. For example, a slave and master node, where the slave node continuously replicates the master node. In case the master node fails, the slave mode is triggered and will stop replicating from the master node, execute a recovery and will become the master node.

ActiveMQ

The components of the Platform communicate with each other through a Message Queue. The Open Smart Grid Platform uses Apache Active Message Queue, which makes asynchronous communication possible between components. The components can register to the queues as consumers. In case a consumer (e.g. a server running a component of the platform) is down, the message will still be consumed by the remaining consumer(s). The Message Brokers can be used as a MasterSlave. In case the Master message broker is down, you get immediate fail-over to the slave without loss of messages.
Redundancy
Performance

This chapter describes the results of a performance test, to give potential users an indication of the system requirements for the platform.

The Platform was tested with the following AWS setup:

Systems:

- **Specifications Component Server**: 2 CPUs, 8 GB RAM
- **Specifications Database Server**: 1 CPU, 2 GB RAM

Setup:

- **Front-end**: 2x Component Server
- **Middle-end**: 2x Component Server
- **Back-end**: 2x Component Server
- **ActiveMQ Front-Middle**: 1x Component Server
- **ActiveMQ Middle-Back**: 1x Component Server
- **Databases**: 1x DB Server

For the test to succeed, two requirements were to be met:

1. Switch 10,000 simulated OSLP devices under 5 minutes.
2. Switch 40,000 simulated OSLP devices under 5 minutes.

The results showed that both tests succeeded.
Technical Overview

Technical Architecture

This chapter gives a more technical overview. It describes each layer of the platform by giving an overview of its packages and the code it contains. Furthermore it describes how a message proceeds through the platform. If you are planning on adding your own Domain Adapter or Protocol Adapter, it will be useful to read this chapter to get a feeling of how the Platform has been built.

A Request through the Platform

The picture below depicts an example of a request (OSLP SetLight request) proceeding through the platform to a device.

- A web request enters the platform at its EndPoint, which in turns calls the RequestService. The RequestService checks if the organisation in the request is authorized, creates the request message and sends it to the MessageSender which in turn puts it on the queue of the Domain Adapter.
- In the Domain Adapter, the incoming message is processed in the MessageProcessor, which in turn calls the RequestService. Here the message is converted to a DTO object. The CoreRequestMessageSender puts the message on the Core Queue.
- The MessageListener in Core receives the message. The DeviceRequestMessageService contains generic functionality such as Authorization, Validation, etc. Once these procedures are completed, the message is routed to the appropriate protocol adapter.
- In the Protocol Adapter the message is received by the MessageListener. It is processed through the MessageProcessor and OslpDeviceService. The request eventually ends up in the OslpChannelHandler, where the actual Protocol Request to the device is made.

For a detailed description of each layer, please take a look at a more detailed description of each layer in this chapter.

Configuration files

The Platform uses property files for certain settings (such as JMS settings, Persistence settings, etc.). These files are stored in property files which can be found in the Config repository on Github. These files are sym linked to /etc/osp/, where the Platform (through reference in context.xml) looks for the property files.
Web Services Layer

Web Services

The Web Services layer contains the web services that are used to communicate with the Platform. The Open Source Smart Grid Platform uses the Simple Object Access Protocol or SOAP to expose its interfaces. The Web Services Adapter receives requests and sends those to the Domain Adapter. An incoming request is converted to a Domain Object, and put on the MessageQueue of the Domain layer. The Web Services layer also has a queue for incoming responses from the Domain adapter.

Each domain of the Platform has its own web services:

**Generic**
- Core - osgp-adapters-core: Contains the Core (common) web services.
- Admin - osgp-adapters-admin: Contains the Admin web services.
- Shared - osgp-adapters-shared: Contains shared endpoints, such as header authorization
- Database - osgp-adapters-db: Contains repositories for persistence.

**Domain**
- Public Lighting - osgp-adapters-publiclighting: Contains the Public Lighting web services.
- Smart Metering - osgp-adapters-smartmetering: Contains the Smart Metering web services.
- Tariff Switching - osgp-adapters-tariffswitching: Contains the Tariff Switching web services.
- Microgrids - osgp-adapters-microgrids: Contains the Micro Grids web services.
- Distribution Automation - osgp-adapters-distributionautomation: Contains the Distribution Automation web services.

For a description of the WSDL's see the Domain Chapter.

General Package structure

A description of the general package structure of a web service component.

**application**
- config: Contains the configuration files for the Component. Uses the property files in /etc/osp/. -- ApplicationContext -- AdapterInitializer -- MessagingConfig -- PersistenceConfig -- WebServiceConfig
- exceptionhandling: Exceptions are defined here.
- mapping: Custom Orika converters.
- services: Contains services used by the domain, such as AdHocManagement. These are called by the end points and convert the request to a Domain Object and put the request on the domain message queue using the JMS classes.

**endpoints**
- EndPoints for the web services: contain a reference to a service that proceeds with handling the request.

**infra**
- jms: contains the JMS classes such as: -- MessageSender(s) -- MessageType -- ResponseMessageFinder

**webapp**

The WSDL and schema definitions can be found under main/webapp/WEB_INF/wsdl.
Domain Layer

Domain Adapters

The Domain Adapters are responsible for receiving requests from the Web Services layer, and delivering them to the Core layer. The Domain Layer mainly contains MessageProcessors and Services for request handling.

The Core/Admin components contains the shared functionality, while the Domain components contain additional domain specific functionality.

At the moment the Platform uses the following Domain Adapters:

Generic
- Core - `osgp-adapter-domain-core`: Contains Core (common) functionality; AdHocManagement, FirmwareManagement, etc.
- Admin - `osgp-adapter-domain-admin`: Contains Admin functionality, e.g. DeviceManagement.

Domain
- Public Lighting - `osgp-adapter-domain-publiclighting`: Contains functionality for the Public Lighting Domain.
- Smart Metering - `osgp-adapter-domain-smartmetering`: Contains functionality for the Smart Metering Domain.
- Tariff Switching - `osgp-adapter-domain-tariffswitching`: Contains functionality for the Tariff Switching Domain.

General Package structure

application
- config: Contains the configuration files for the Component. Uses the property files in /etc/osp/. -- ApplicationContext -- DomainAdapterInitializer -- MessagingConfig -- PersistenceConfig
- mapping: Custom Orika converters for mapping to/from DomainObjects/DTO Objects.
- services: Contains most of the domain logic, related to the specific services of the adapter. The service classes converts DTO objects to Domain objects (or vice versa), and put the request on the Core queue through the JMS classes.

infra
- `jms.core`: Inbound/outbound messages from/to the Core layer. This package contains Messages, MessageListeners, MessageSenders and MessageProcessors for sending requests to the Core Queue, or receiving and processing responses from Core.
- `jms.ws`: Inbound/outbound messages from/to the web services layer. This package contains Messages, MessageListeners, MessageSenders and MessageProcessors for sending requests to the Web Services Queue, or receiving and processing responses from the web services layer.
Core Layer

The Core layer of the Open Source Grid Platform is responsible for Validation, Translation, Authorisation and Routing of request messages. It also contains all the Domain Objects.

The core layer consists of two components:

- **osgp-domain-core**: Shared Domain objects, services, repositories, etc. These classes are used throughout the entire platform.
- **osgp-core**: Logic for routing domain requests, scheduling, retrying, etc.

**General Package structure: osgp-domain-core**

- entities: Defines the entities used for persistence.
- exceptions: Domain specific exceptions reside here.
- repositories: Repositories that contain logic for persisting entities.
- services: Domain services that reference a repository.
- specifications: Interfaces that define specifications for Devices and Events.
- validations: Validators and constraints.
- valueobjects: Definitions of the Domain Objects.

**General Package structure: osgp-core**

**application**

- config: Contains the configuration files for the Component. Uses the property files in /etc/osp/. --ApplicationContext -- OsgpCoreInitializer -- DomainMessagingConfig -- PersistenceConfig -- ProtocolMessagingConfig -- SchedulingConfig
- services: Services that process device requests/responses. Checks for authorization, and if the request is supported by the platform, it will be routed to the appropriate protocol adapter.
- tasks: Contains task scheduler logic.

**domain.model**

These packages contain interfaces for the Services.

- domain: Interfaces for the Domain services.
- protocol: Interfaces for the Protocol services.

**infra.jms**

- domain: Contains Messages, MessageListeners and MessageProcessors for Domain related messaging.
Protocol Layer

Protocol Adapters

The Protocol Adapters are responsible for the actual communication to and from a device. They usually operate in a certain domain, and might use common/generic functions from the platform.

The Open Smart Grid Platform currently has the following protocol adapters:

- Protocol-Adapter-DLMS: Smart Metering
- Protocol-Adapter-IEC61850: Public Lighting, Micro Grids and Distribution Automation
- Protocol-Adapter-OSLP: Public Lighting and Micro Grids

General package structure

**application**

- config: Contains the configuration files for the Component. Uses the property files in /etc/osp/. -- ApplicationContext
- MessagingConfig
- OsgpProtocolAdapterOslpInitializer
- OslpConfig
- OslpPersistenceConfig
- services: Contains the (functional) services that control communication from (and to) the device. Also persists requests and responses, and deals with security. Actual communication is done through the classes in the infra package.

**device**

Contains the Protocol Adapter Objects used for the Protocol Adapter.

- requests: Objects representing the requests that are supported by this adapter.
- responses: Objects representing the responses that are supported by this adapter.

**domain**

- entities: Entities used for persistence.
- repositories: Repositories used for persistence.

**exceptions**

Contains the exceptions that might be thrown while communication with a device, e.g. ValidationException, MessageRejectedException, etc.

**infra**

This package contains all the code for communication through JMS (Platform) and Networking (Device).

- messaging: Contains the JMS Messages, MessageListeners, MessageSenders and MessageProcessors.
- networking: Contains the classes that are responsible for connecting to a device using a certain protocol.

**services**

Services used to check the request status.
Technology Stack

Tools and technology used

Platform

- **Apache ActiveMQ**: Open source messaging server, used to relay messages between components of the open smart grid platform. ActiveMQ is an open source message broker written in Java and a full Java Message Service (JMS) client. It provides “Enterprise Features” which in this case means fostering the communication from more than one client or server.
- **Apache HTTP server**: Web server, used as front for Apache Tomcat.
- **Apache Tomcat**: Provides a "pure Java" servlet container for Java code to run in.
- **pgAdmin-III**: PostgreSQL administration and management tools.
- **Protobuf (Google Protocol Buffers)**: A language-neutral, platform-neutral, extensible way of serializing structured data for use in communications protocols, data storage, and more.
- **Flyway**: Agile database migration framework for Java
- **HikariCP**: JDBC connection pool
- **Hibernate**: Object/Relational mapping
- **Netty**: Network application framework for protocol servers & clients
- **OpenMUC**: Library implementing the IEC61850 and DLMS/COSEM communication standard
- **Orika**: Java bean mapping
- **Spring**: Application development framework. Several Spring libraries are used, including Spring Data, Spring Security and Spring WS.
- **Puppet**: Application for Automatically delivering, operating and securing your infrastructure

Development

- **Bower**: Package manager for Javascript packages. Web applications consist of various components; frameworks, libraries, assets, utilities, and rainbows. Bower manages all these things for you.
- **Eclipse**: IDE for developing software.
- **FileZilla**: FTP application.
- **Git**: Version control system.
- **NodeJS**: Tooling suite with various Javascript tools.
- **NPM**: Package manager for the NodeJS Javascript applications.
- **Putty**: A free and open-source terminal emulator, serial console and network file transfer application.
- **Vim**: Source code editor.
- **Apache Maven**: Software project management tool.
- **GIT & GitHub**: Source code management.

Testing & QA

- **Apache JMeter**: Application designed to load test functional behaviour and measure performance.
- **Cucumber**: automated acceptance testing framework.
- **Gherkin**: DSL for acceptence testing framework.
- **Sonarqube**: Quality management platform.
- **SoapUI**: Functional testing tool for testing web services.
- **JUnit**: Unit testing.
- **Mockito**: A Mock framework for Unit testing.

The following table presents an overview of the components and the most important technical choices per component.

<table>
<thead>
<tr>
<th>Component (not open source)</th>
<th>Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSGP Management application</td>
<td>Java, Spring Framework, Spring MVC</td>
</tr>
<tr>
<td>Net-Management application</td>
<td>Java, Spring Framework, JAX-RS, AngularJS</td>
</tr>
<tr>
<td>Liander Installatie application</td>
<td>Java, Spring Framework, JAX-RS, AngularJS</td>
</tr>
</tbody>
</table>
Use cases

The open smart grid platform use-cases are strongly related to the open smart grid platform domains. Up-to-date information on use-cases can be found on the Grid eXchange Fabric website.
General User's Guide

CHAPTER 2. User's Guide

Sys Admins who are tasked with keeping the Open Source Grid Platform running on an environment, can find some information in this chapter.

Get Started

To get started with Open Source Grid Platform, please read our Introduction. The Introduction offers an excellent overview of the Platform and its features.

A next step could be to have a look at the WSDL's to understand which functions are present per functional domain. Depending on the functional domain one is interested in, one could also have a look at the Protocol Adapter and device simulator for the domain.

Installation

If a full installation is desired, have a look at our Installation Guide. This can be used to setup a development environment which can be used to start the Platform and run it. Installation on one or several servers can be derived from the steps within the Installation Script.
Installation Guide

Installation Script

To get started quickly, a Vagrant Installation Guide has been created and a guide for Manual Installation.

The goal of the installation manual is to control a simulated OSLP device through the Platform. Below, is a summary of all steps involved. See the next chapters for a detailed guide with screenshots. Please follow the steps carefully.

A summary of the steps involved:

- Creating a virtual machine using Vagrant and Virtual Box
- Run the puppet script (part of the Vagrant installation), or complete the steps manually (Manual installation)
- Importing Maven Projects into Eclipse
- Creating an Apache Tomcat9 Server
- Setting Up Apache Tomcat9 Server Context
- Deploying all open smart grid platform components to an Apache Tomcat9 Server
- Starting Apache ActiveMQ
- Starting Apache Tomcat9 Server
- Creating the 'test-org' organization in the Database
- Setting up SoapUI
- First SOAP Requests to add a device to the open smart grid platform
- Opening Device Simulator to add a device
- Registering a device
- Using 'SetLight' SOAP Request to switch the light on
Installation

Open Smart Grid Platform Installation

To install the platform you can use one of the following procedures.

1. The Vagrant Installation. This procedure creates and installs a complete image with the Open Smart Grid Platform pre-installed, including all the tools such as Maven, Eclipse, SoapUI etc.
2. The Manual Installation. Follow this guide if you want to install the Open Smart Grid Platform yourself.
Vagrant

Installation

This document describes the automatic installation procedure for your Open Smart Grid Platform development environment.

Manual installation

If you would like to follow the manual installation procedure, please proceed to the Manual Installation Chapter.

Overview

Creating a Virtual Machine using VirtualBox and Vagrant

To improve the usability of the Installation process, a Vagrant file and some puppet scripts are used to automatically set-up an virtual Open Smart Grid Platform development environment. The following steps will describe how to install VirtualBox, Vagrant and kick off the procedure by running the `vagrant up` command.

System Requirements

The following system requirements are recommended:

- Core i5/i7 ~2.5GHz Dual Core, Quad Core recommended
- At least 6 GB RAM, 8 GB RAM recommended
- At least 20 GB free space, 50 GB free space recommended

The installation procedure has been tested on Windows 7, Windows 10, MacOS, Ubuntu 14.04 and Ubuntu 16.04.

Install Vagrant and VirtualBox

Start by downloading VirtualBox by going to https://www.virtualbox.org/wiki/Downloads And follow the installation steps.

note: If you already have VirtualBox, make sure it is at least version 5.1.32
note: Check whether Virtualbox stores the images on a drive with enough free space. (Open Oracle VM VirtualBox Manager -> Preferences -> General -> Default Machine Folder).

Now download and install Vagrant. Vagrant is available at the following URL: https://www.vagrantup.com/downloads.html
note: If you already have Vagrant, make sure it is at least version 2.1.1 Complete the installation and restart your PC.

note: If you did a fresh install of Vagrant and already had a command prompt open, make sure you close this command prompt and open it again.

Tip

- Remember to enable Virtualization in your system BIOS.
- Also disable Hyper-V in Windows (can be found in Windows Features)

Download and run the Vagrant file

First create a new directory (for example: D:\My Vagrant Images\OSGP Development)

Browse to https://github.com/OSGP/Config/tree/development/vagrant and save the png image and Vagrantfile files in your newly created directory.
Note

- Make sure that the file is named like this: Vagrantfile without an extension!
- If the file has an extension (for example .txt) you can rename the file using the following console command.
  MacOS/Linux:
  ```shell
  mv Vagrantfile.txt Vagrantfile
  ```
  Windows:
  ```shell
  move Vagrantfile.txt Vagrantfile
  ```

Now open a Command Prompt and navigate to the newly created directory where you just put the files. Make sure that you run the Command Prompt using administrator rights.

**note:** When you open the Vagrantfile you see that default the image is configured to run in virtualbox with 2 cpu cores and 8192 MB of RAM. If you need to you can change this to more or less cpu cores and RAM, but it is recommended to use the provided settings.

Run the following command: `vagrant up`

**note:** In case of an error bad uri Images/OSGP Development/hashicorp/cxtlabs/vagrant-ubuntu-16.04-mate then use the following command;

- `vagrant destroy`
- `vagrant box add cxtlabs/vagrant-ubuntu-16.04-mate`
- `vagrant up`

**note:** In case of an error complaining about not being able to resolve a URL (for instance to github.com) then try using a different internet connection not behind a proxy.
Vagrant will now automatically download an Ubuntu image (~2.6 Gb), create a virtualbox image from it and run the installation puppet script when finished. This might take a while, depending on your internet speed. After some time (while the script is still running) you will notice that a window with an Ubuntu Virtual Machine pops-up. Don't log in yet, wait until the script in the Console is finished.

```
$ cd "My Vagrant Images"
$ cd "OSGP Development"
$ cd OSGP Development:
$ Volume in drive D is Data
$ Volume Serial Number is E8BF-BB8D
$ Directory of D:

15/10/2016 20:18 <DIR>
13/10/2016 20:18 <DIR>
11/10/2016 20:17 2,001,174 OSGP-solar.png
10/10/2016 20:17 1,910 Vagrantfile
2 File(s) 2,901,246 bytes
2 Dir(s) 291,742,949,378 bytes free
$ cd OSGP Development:
$ vagrant up
```

Tip

Installation
- If the script fails for some reason (e.g., Errors in the console such as time outs during downloading), you can retry the procedure by running the following command: `vagrant destroy && vagrant up`.

Now that the script has ran its course, it will automatically log in on the Ubuntu virtual machine as user `dev`.

**Note:** For some actions, like `sudo`, you will have to enter the password of user `dev`. The password for user `dev` is `dev`.

**Optimize your Open Smart Grid Platform Development virtualbox image.**

At this point you also can adjust the virtualbox settings like cpus and memory size. If you don't want to adjust this proceed to Chapter 2.1.2. Platform Setup.

If you do want to update the virtualbox settings for this image, shut down the image first:
Once the machine has been Shut Down, open VirtualBox and right click on the new virtual machine (called "OSGP Development") and select Settings. Go to System and increase the Base Memory of the system to at least 6144 MB (6 GB) (or the maximum recommended (in green) amount for your system).

Now go to the Processor Tab and increase the amount of Processors to the maximum recommended (in green) amount.
Close the Settings window and Start the Virtual Machine again. Once it is booted, you should be automatically logged in as the 'dev' user.

**Post actions**

In order to use git correctly you need to execute the following commands in a terminal:

```
git config --global user.name "your full name"
git config --global user.email yourEmailAddress
```

**Recap**

You just created a virtual machine running Ubuntu with pre-installed tooling. Proceed with Platform Setup of the guide describing how to set-up the open smart grid platform.
Manual Setup

Manual Installation

This chapter describes the steps for a manual installation (e.g., not using the vagrant script and puppet scripts). This chapter is for developers who would like to have more control over the installation procedure.

With the increased control comes increased risks of things not working with the versions or configuration of the software involved with the OSGP environment. If you run into issues, you may find clues in the puppet scripts about versions and modifications to the configuration of installed software.

Note

- Skip this chapter if you followed the Vagrant installation! You can continue with next chapter: Setup the Open Smart grid Platform

Operating System

The Open Smart Grid Platform runs on a Linux environment. It is recommended to set up a machine running Ubuntu.

Software and tools

The Open Smart Grid platform needs the following software and tools installed/downloaded:

- Java 8 openjdk-8
- PostgreSQL and pgAdmin 3
- Git
- Maven
- ActiveMQ
- Tomcat
- Apache HTTPD
- SoapUI
- Eclipse IDE for Java EE Developers
- Google Protocol Buffers: protobuf-compiler, libprotoc7 and libprotobuf7
- PostgreSQL JDBC driver

Settings

User

It is recommended to create a 'dev' user, because some scripts contain hard coded references to this 'dev' user. It is possible to skip this step, but then some of the scripts will have to be adjusted manually.

Tomcat

- Place the PostgreSQL JDBC driver jar in the Tomcat lib directory.
- Change permissions of Tomcat Config files to 644 in the Tomcat conf directory.

Apache HTTPD

- Enable mod_ssl by running the following command:
  
a2enmod ssl

- Enable proxy_ajp by running the following command:
  
a2enmod proxy_ajp

Java

- Make sure the JAVA_HOME var is set, and points to openjdk-8.

Cloning Sources

Clone the following repo's, it is recommended to create a Sources/OSGP directory in /home/dev/ since some scripts contain hard coded references to those folders.

```
git clone https://github.com/OSGP/Config.git /home/dev/Sources/OSGP/Config
git clone https://github.com/OSGP/open-smart-grid-platform.git /home/dev/Sources/OSGP/open-grid
git clone https://github.com/OSGP/Documentation.git /home/dev/Sources/OSGP/Documentation
```
Make sure you are on the development branch (default).

**Creating directories and symlinks**

Create the following directories:

- `/var/log/osp/logs`
- `/etc/osp/`

Make the dev user (or equivalent) the owner of the log directory with rwx permission. Give the other users read and execute permission.

Execute the script `/home/dev/Sources/OSGP/Config/scripts/create-symlinks.sh`

*Note* This script uses hard coded references to `/home/dev/Sources/OSGP/*`, if you used a different user, please edit the script before executing it.

The script will make symlinks to certificates, to Apache HTTP server configuration and copy configuration settings as samples to locations where these properties may be overridden.

**Initiating the Database**

To create the database run the following command (Change `/home/dev/` in case of no dev user)

```
sudo -u postgres /usr/bin/psql -p 5432 -f /home/dev/Sources/OSGP/Config/sql/create-users-and-databases.sql
```

And create a backup of the `pg_hba.conf` file (modify if your version of PostgreSQL is different)

```
```

Finally, reload the postgresql service:

```
service postgresql reload
```

**Set up the Open Smart Grid Platform**

Continue with setting up the Open Smart Grid Platform by following the [Set up the Open Smart Grid Platform Guide](#)
Platform Setup

Setting Up the Open Smart Grid Platform Development environment

This chapter describes all the steps needed to finalize the open smart grid platform development environment.

Lombok

The platform uses Lombok annotations to generate extra Java methods. Without Lombok the project is not imported correctly by Maven and will not run. If you used the Vagrant installation method it should already be installed. To check if Lombok is properly installed go Help > About Eclipse IDE and scroll down. Here you will see: Lombok <version> "<version name>" is installed. https://projectlombok.org/. If not follow this guide to install Lombok.

Importing Maven Projects into Eclipse

Open Eclipse by clicking the shortcut on the Desktop and import the projects.

Go to File -> Import -> Existing Maven Projects, browse to folder /home/dev/Sources/OSGP

Import the projects from location /home/dev/Sources/OSGP/open-smart-grid-platform.

Creating an Apache Tomcat Server

In Eclipse go to Window -> Open Perspective -> Debug
In the 'Debug' perspective, go to the 'Servers' view and add a new Apache Tomcat server. Tomcat is available in the folder `/home/dev/Tools/tomcat` (or in another location if you didn't set up a VM using Vagrant, the latest version usually works fine).

Click on Next
Click on Finish
After adding the server, double click on the Tomcat server in the 'Servers' view and set the following configuration: under 'Timeouts' set 'Start' to 600 and 'Stop' to 30.
Click on 'Open launch configuration', click on the 'Arguments' tab and add the following at the end of the 'VM arguments':

-Xms512m -Xmx2048m -Xss512k -XX:MaxMetaspaceSize=1024m -XX:+CMSClassUnloadingEnabled -XX:+UseConcMarkSweepGC -Dcom.sun.management.jmxremote=true
Setting Up Apache Tomcat Server Context

All modules contain their own context.xml. In the module specific context.xml are the environment variables defined where the global and module specific configuration files are located. Default they will point to a location in /etc/osp/.

If you want to deviate from this, you might set up the context.xml in Tomcat to be able to redirect in one file to different locations. This is optional and not required. In order to use a custom context.xml, copy the entries in /home/dev/Sources/OSGP/Config/tomcat/context.xml.sample to the Tomcat context.xml in the eclipse Servers folder, to map configuration file names to file paths.
Deploying all Open Smart Grid Platform components to Apache Tomcat Server

Continue by adding the Maven Projects to the Tomcat server by right clicking on the Tomcat server and choosing 'Add and Remove'. Select all available resources, except for osgp-protocol-simulator-61850 (which is for advanced use and requires additional configuration), then click the 'Add' button.
At this point, eclipse's auto-build should have built the projects, and the Tomcat server has been setup.

**Starting Apache ActiveMQ**

Continue with starting Apache ActiveMQ. If you installed an environment as described with Vagrant, you can double click the ActiveMQ shortcut on the desktop.

Alternatively you can open a terminal and run the executable manually by using the following command: (the executable can be found in the folder `/home/dev/Tools/activemq/bin/linux-x86-64`)

```
sudo ./activemq console
```

This starts ActiveMQ as a terminal process (this way, ActiveMQ doesn't detach from the terminal and starts running as a daemon).
Starting Apache Tomcat Server

With ActiveMQ running, the Tomcat server can be started. Go to Eclipse, go to the Servers tab in the Debug view, and right click on the Tomcat server and select 'Start'.

*note*: In case of an error starting up for the very first time, try and start up only the module: 'osgp.core' first. This makes sure the database scripts are executed.

Probe

This is an optional program that shows the status of the Tomcat resources in real time. To install Probe you can follow this guide. Note that you need to add the Tomcat users in the guide's Security part in the tomcat-users.xml in your Eclipse environment. To download the war file go to: [https://github.com/psi-probe/psi-probe/releases](https://github.com/psi-probe/psi-probe/releases). You need to copy the war file.
Starting pgAdmin III and Connect to PostgreSQL

Open pgAdmin III and configure a connection: choose the 'Add a connection to a server.' and fill out the fields using:

- Host: localhost
- Port: 5432
- Username: osp_admin
- Password: 1234
Creating the 'test-org' Organization (in database osgp-core)

Run the script in /home/dev/Sources/OSGP/Config/sql/create-test-org.sql to insert 'test-org' organization into the organisation table of the osgp_core database.

```
psql -U osp_admin -h localhost -d osgp_core -f /home/dev/Sources/OSGP/Config/sql/create-test-org.sql
```

If asked for a password, enter 1234

Go back to PgAdmin III, expand servers, select localhost -> databases -> osgp_core -> Schemas -> public -> Tables. Right click the organisation table and select to view data for the top 100 rows. Confirm that the test-org organisation has been added to the Database.
Now that everything has been set up, continue to the next chapter to start testing the Platform by sending it some requests.
Test the Platform

Testing the Open Smart Grid Platform

There are two procedures for testing the Open Smart Grid Platform.

1. **SoapUI**. Create and send Soap requests to the Platform to manage a (simulated) light.
2. **PublicLighting Demo App**. Use the Demo App to send requests to the Platform to manage a (simulated) device.
Using SoapUI

Testing the platform

This chapter will describe the steps needed to test the Open Smart Grid Platform.

Setting Up SoapUI

Start SoapUI by double clicking the shortcut on the Desktop or run it manually by typing the following command in a terminal:

```
/home/dev/Tools/SoapUI/bin/soapui.sh
```

Go to File -> Preferences -> SSL Settings, and browse for the KeyStore to

```
/home/dev/Sources/OSGP/Config/certificates/osgp-ca/certs/test-org.pfx
```

and fill out the password (the password is 1234).

Go to WSDL Settings and check ‘Generate Example Values in New Requests’ and ‘Generate Comments with Type Information in New Requests’.
Add the SoapUI projects to SoapUI

There are several SoapUI project prepared, see /home/dev/Sources/OSGP/Config/soapui/. Import all SoapUI projects present in the folder mentioned above. Below, 2 projects are shown as examples.

Adding the 'Admin' Soap project

Import the 'admin' project by clicking File -> Import project. Browse to /home/dev/Sources/OSGP/Config/soapui/, select 'admin-soapui-project.xml' and click open.

Alternatively you can create the 'admin' project yourself by following the steps below:

- Create a new SOAP Project and call it 'admin'
- Open the Project View by double-clicking on the 'admin' project. Go to 'WS-Security Configurations' and select the 'Keystores' Tab. Click on the '+' to add the test-org.pfx in
Fill out the password (1234) and click Ok and close the Project View window.

Right click the 'admin' project and choose 'Add WSDL'. Enter the following URL in the WSDL Location field:

```
/home/dev/Sources/OSGP/open-smart-grid-platform/osgp/shared/osgp-ws-admin/src/main/resources/DeviceManagement.wsdl
```

Make sure the box 'Create sample requests for all operations' is checked, and click OK.

Adding the 'Public Lighting' Soap project.

Import the 'public-lighting' project by clicking File -> Import project. Browse to

```
/home/dev/Sources/OSGP/Config/soapui/
```

Alternatively you can create the 'public-lighting' project yourself by following the steps below:

Create another new SOAP Project and call it 'public-lighting'
Open the Project View by double-clicking on the 'public-lighting' project. Go to 'WS-Security Configurations' and select the 'Keystores' Tab. Click on the '+' to add the test-org.pfx in /home/dev/Sources/OSGP/Config/certificates/osgp-ca/certs/

Fill out the password (1234) and click Ok and close the Project View window.

Right click the 'public-lighting' project and choose 'Add WSDL'. Enter the following URL in the WSDL Location field: /home/dev/Sources/OSGP/open-smart-grid-platform/osgp/shared/osgp-ws-publiclighting/src

Make sure the box 'Create sample requests for all operations' is checked, and click OK.
First SOAP requests to add a device to the open smart grid platform

Before sending the request, the test-org.pfx should be added as SSL Keystore: Go to the properties interface for the request (bottom left of the screen, after selecting 'Request 1' under UpdateKey in the 'admin' project), and choose test-org.pfx from the drop-down box.

Note

- This has to be done for each request!

An SSLD needs to be added to the platform, as well as a manufacturer and a public key for the SSLD. A couple of steps need to be performed to realize this.

1. Add manufacturer
2. Add device model
3. Add SSLD
4. Setup a protocol for the SSLD to use
5. Set the public key for the SSLD (in case of OSLP)

Test the Platform
The AddManufacturer function adds a new manufacturer to OSGP. All devices are coupled to a manufacturer.

```xml
  xmlns:ns1="http://www.opensmartgridplatform.org/schemas/energyinformation/2014/10">
  <soapenv:Header>
    <ns:ApplicationName>APPLICATION_NAME</ns:ApplicationName>
    <ns:UserName>USER_NAME</ns:UserName>
    <ns:OrganisationIdentification>test-org</ns:OrganisationIdentification>
  </soapenv:Header>
  <ns1:AddManufacturerRequest>
    <ns1:Manufacturer>
      <ns1:Id>3</ns1:Id>
      <ns1:Code>MAN</ns1:Code>
      <ns1:Name>Manufacturer01</ns1:Name>
      <ns1:UsePrefix>false</ns1:UsePrefix>
    </ns1:Manufacturer>
  </ns1:AddManufacturerRequest>
</soapenv:Envelope>
```

The AddDeviceModel function adds a new device model to OSGP. All devices are coupled to a device model.

```xml
  xmlns:ns1="http://www.opensmartgridplatform.org/schemas/energyinformation/2014/10">
  <soapenv:Header>
    <ns:ApplicationName>APPLICATION_NAME</ns:ApplicationName>
    <ns:UserName>USER_NAME</ns:UserName>
    <ns:OrganisationIdentification>test-org</ns:OrganisationIdentification>
  </soapenv:Header>
  <ns1:AddDeviceModelRequest>
    <ns1:DeviceModel>
      <ns1:Manufacturer>MAN</ns1:Manufacturer>
      <ns1:ModelCode>MOD01</ns1:ModelCode>
      <ns1:Description>Device model MOD01.</ns1:Description>
    </ns1:DeviceModel>
  </ns1:AddDeviceModelRequest>
</soapenv:Envelope>
```

The AddDevice function adds a new SSLD to OSGP. The device is coupled to a device model and a manufacturer.

```xml
  xmlns:ns1="http://www.opensmartgridplatform.org/schemas/energyinformation/2014/10">
  <soapenv:Header>
    <ns:ApplicationName>APPLICATION_NAME</ns:ApplicationName>
    <ns:UserName>USER_NAME</ns:UserName>
    <ns:OrganisationIdentification>test-org</ns:OrganisationIdentification>
  </soapenv:Header>
  <ns1:AddDeviceRequest>
    <ns1:Device>
      <ns1:DeviceIdentification>SSLD_000-00-01</ns1:DeviceIdentification>
      <ns1:Owner>test-org</ns1:Owner>
      <ns1:Activated>false</ns1:Activated>
      <ns1:HasSchedule>false</ns1:HasSchedule>
      <ns1:PublicKeyPresent>false</ns1:PublicKeyPresent>
      <ns1:DeviceModel>
        <ns1:Manufacturer>MAN</ns1:Manufacturer>
        <ns1:ModelCode>MOD01</ns1:ModelCode>
      </ns1:DeviceModel>
    </ns1:Device>
  </ns1:AddDeviceRequest>
</soapenv:Envelope>
```
The function UpdateDeviceProtocol sets a protocol for a device.

```xml
xmlns:ApplicationName=http://www.example.com/AppName
xmlns:UserName=http://www.example.com/UserName
xmlns:OrganisationIdentification=http://www.example.com/OrgIdentification"
xmlns:DeviceIdentification=http://www.example.com/DeviceIdentification"
xmlns:ProtocolInfo=http://www.example.com/ProtocolInfo"
xmlns:Id=http://www.example.com/Id"
xmlns:Protocol=http=http://www.example.com/Protocol"
xmlns:ProtocolInfoId=http://www.example.com/ProtocolInfoId">

<soapenv:Header>

<ns:ApplicationName>APPLICATION_NAME</ns:ApplicationName>
<ns:UserName>USER_NAME</ns:UserName>
<ns:OrganisationIdentification>test-org</ns:OrganisationIdentification>
</soapenv:Header>

<ns1:AddDeviceRequest>
</ns1:AddDeviceRequest>
</soapenv:Body>
</soapenv:Envelope>
```

The UpdateDeviceProtocolRequest sets a protocol for a device.

```xml
xmlns:ApplicationName=http://www.example.com/AppName
xmlns:UserName=http://www.example.com/UserName
xmlns:OrganisationIdentification=http://www.example.com/OrgIdentification"
xmlns:DeviceIdentification=http://www.example.com/DeviceIdentification"
xmlns:ProtocolInfo=http=http://www.example.com/ProtocolInfo"
xmlns:Id=http=http://www.example.com/Id"
xmlns:Protocol=http=http://www.example.com/Protocol"
xmlns:ProtocolInfoId=http=http://www.example.com/ProtocolInfoId">

<soapenv:Header>

<ns:ApplicationName>APPLICATION_NAME</ns:ApplicationName>
<ns:UserName>USER_NAME</ns:UserName>
<ns:OrganisationIdentification>test-org</ns:OrganisationIdentification>
</soapenv:Header>

<ns1:AddDeviceRequest>
</ns1:AddDeviceRequest>
</soapenv:Body>
</soapenv:Envelope>
```

The UpdateKey function of the admin webservice sets a public key for a device. Double click 'Request 1' under UpdateKey in the 'admin' project. Add the following request:

```xml
xmlns:ApplicationName=http://www.example.com/AppName
xmlns:UserName=http://www.example.com/UserName
xmlns:OrganisationIdentification=http://www.example.com/OrgIdentification"
xmlns:DeviceIdentification=http=http://www.example.com/DeviceIdentification"
xmlns:ProtocolInfoId=http=http://www.example.com/ProtocolInfoId">

<soapenv:Header>

<ns:ApplicationName>APPLICATION_NAME</ns:ApplicationName>
<ns:UserName>USER_NAME</ns:UserName>
<ns:OrganisationIdentification>test-org</ns:OrganisationIdentification>
</soapenv:Header>

<ns1:UpdateKeyRequest>
</ns1:UpdateKeyRequest>
</soapenv:Body>
</soapenv:Envelope>
```

The UpdateKeyRequest sets a public key for a device. Double click 'Request 1' under UpdateKey in the 'admin' project. Add the following request:

```xml
xmlns:ApplicationName=http://www.example.com/AppName
xmlns:UserName=http://www.example.com/UserName
xmlns:OrganisationIdentification=http://www.example.com/OrgIdentification"
xmlns:DeviceIdentification=http=http://www.example.com/DeviceIdentification"
xmlns:ProtocolInfoId=http=http://www.example.com/ProtocolInfoId">

<soapenv:Header>

<ns:ApplicationName>APPLICATION_NAME</ns:ApplicationName>
<ns:UserName>USER_NAME</ns:UserName>
<ns:OrganisationIdentification>test-org</ns:OrganisationIdentification>
</soapenv:Header>

<ns1:UpdateKeyRequest>
</ns1:UpdateKeyRequest>
</soapenv:Body>
</soapenv:Envelope>
```
After the SSLD has been added, let's see if the function FindAllDevices shows the SSLD. Continue with the FindAllDevices request from the public-lighting project. Since this is not the same project, we have to change the endpoint; in this case in https://localhost:443/osgp-adapter-ws-publiclighting/publiclighting/adHocManagementService/. Do not forget to set the SSL keystore in the Request Properties. Use the following parameters in the request:

     xmlns:ns1="http://www.opensmartgridplatform.org/schemas/deviceManagement/2014/12"
     xmlns:ns2="http://www.opensmartgridplatform.org/schemas/deviceManagement/2014/12">
     <soapenv:Header>
         <ns:DeviceIdentification>
             <ns:DeviceIdentificationId ns1:DeviceId="A"/>
             <ns:DeviceType ns1:DeviceType="PublicLightingDevice"/>
             <ns:DeviceSubType ns1:DeviceSubType="PublicLightingDeviceSubType"/>
             <ns:DeviceSeries ns1:DeviceSeries="PublicLightingDeviceSeries"/>
             <ns:DeviceModel ns1:DeviceModel="PublicLightingDeviceModel"/>
             <ns:DeviceSerialNumber ns1:DeviceSerialNumber="PublicLightingDeviceSerialNumber"/>
             <ns:DeviceOwner ns1:DeviceOwner="PublicLightingDeviceOwner"/>
             <ns:DeviceManufacturer ns1:DeviceManufacturer="PublicLightingDeviceManufacturer"/>
             <ns:DeviceLifeCycle ns1:DeviceLifeCycle="PublicLightingDeviceLifeCycle"/>
             <ns:DeviceLocation ns1:DeviceLocation="PublicLightingDeviceLocation"/>
         </ns:DeviceIdentification>
         
         <ns:ProtocolInfo>
             <ns:Protocol ns1:Protocol="OsgpAdapterPUBLICLIGHTING"/>
             <ns:ProtocolVersion ns1:ProtocolVersion="PUBLICLIGHTING1.0"/>
         </ns:ProtocolInfo>
     
         <ns:DeviceManagementPortSoap>
             <ns:DeviceManagementPortSoapId ns1:DeviceManagementPortSoapId="PUBLICLIGHTING1.0"/>
             <ns:DeviceManagementPortSoapVersion ns1:DeviceManagementPortSoapVersion="PUBLICLIGHTING1.0"/>
         </ns:DeviceManagementPortSoap>
     </soapenv:Header>

     <soapenv:Body>
         
             <ns1:FindDeviceRequest>
                 
                     <ns1:DeviceIdentificationId ns1:DeviceIdentificationId="A"/>
                    </ns1:FindDeviceRequest>
     
         </soapenv:Body>

     </soapenv:Envelope>
After the request has been submitted, the response should include the SSLD device with ID SSLD_000_00_01

Opening Device Simulator to Add a Device

In order to be able to use the SSLD-000-00-01 Device, the device needs to be simulated in the Device Simulator. To do this we have to create it. In the Firefox Browser, open the Device Simulator by going to the following URL:

https://localhost/web-device-simulator/devices

If you encounter an Untrusted Connection page, go to 'I Understand the Risks' -> Add Exception.. -> Confirm Security Exception

Click Add Device
Fill out the fields like this:

- **Device Identification:** SSLD_000-00-01
- **IP Address:** 127.0.0.1
- **Device Type:** SSLD
- **Protocol:** OSLP ELSTER

Click Create Device

You should return to the Devices screen and see the message "Device with identification SSLD_000-00-01 was created."
Registering a Device

Now click on the newly created device and click the 'Register Device' button. After a while the message "Device identification with identification SSLD_000-00-01 was registered at XXXXXXXX" appears.

Then click the 'Confirm Registration' button. The message should read: "Device with identification SSLD_000-00-01 was confirmed to be registered."
Using 'SetLight' SOAP Request to Switch the Light On

Now that the Device is known in the platform, and simulated in the Device-Simulator, the device can be used. Let's switch on the Light. Using SoapUI, click on Request 1 under SetLight at the public-lighting project. Set the following parameters in the request (And do not forget to set the Keystore in the request properties):

```xml

<soapenv:Header>
  <ns:ApplicationName>APPLICATION_NAME</ns:ApplicationName>
  <ns:UserName>USER_NAME</ns:UserName>
  <ns:OrganisationIdentification>test-org</ns:OrganisationIdentification>
</soapenv:Header>

<soapenv:Body>
  <ns1:SetLightRequest>
    <!--type: Identification-->
    <ns1:DeviceIdentification>SSLD_000-00-01</ns1:DeviceIdentification>
    <!--1 to 6 repetitions:-->
    <ns1:LightValue>
      <!--Optional:-->
      <!--anonymous type-->
      <ns1:Index>0</ns1:Index>
      <!--type: boolean-->
      <ns1:On>true</ns1:On>
      <!--Optional:-->
      <!--anonymous type-->
      <ns1:DimValue>100</ns1:DimValue>
    </ns1:LightValue>
  </ns1:SetLightRequest>
</soapenv:Body>
</soapenv:Envelope>
```

Submit the request. Take note of the CorrelationUid in the response. You can use this Id in another request to ask the server for the status of this request.
In the home screen of the OSLP device simulator, the lightbulb should light up for SSLD_000-00-01. This means that the request succeeded.

The last request concerns the response from the previous SetLight request. In SoapUi open Request 1 under 'GetSetLightResponse' in the 'public-lighting' project. Set the following parameters in the request (And the keystore in the request properties). Make sure to replace the CorrelationUid with the value from the response from the SetLight request.

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns1:SetLightAsyncRequest>
      <!--type: CorrelationUid-->
      <ns:CorrelationUid>test-org|||SSLD_000-00-01|||20160721083641940</ns:CorrelationUid>
      <!--type: Identification-->
      <ns:DeviceId>SSLD_000-00-01</ns:DeviceId>
    </ns1:SetLightAsyncRequest>
  </soapenv:Body>
</soapenv:Envelope>
```
Note

- Do not forget to set the CorrelationUid to value in the response you received from the setLight request.

The server replied Ok, indicating that the SetLight request has been processed successfully.

This step also concludes the installation manual.
Using the Demo App

Testing the platform using the Web Demo

This chapter will describe the steps needed to test the Open Smart Grid Platform using the Web Demo Application.

If you followed the steps from the ‘Platform Setup’ then you can skip the following installation steps, because the Web Demo will already be available.

Installing the Web Demo

- Open Eclipse and import the following folder as Maven project /home/dev/Sources/OSGP/open-smart-grid-platform/public-lighting-demo-app/
- Add the PublicLightingDemoApp components to the Tomcat Server.
- (Re)Start the Tomcat Server.

Creating a device

To access the Demo App go to the following URL: https://localhost/web-demo-app/

If you encounter an Untrusted Connection page, go to ‘I Understand the Risks’ -> Add Exception. -> Confirm Security Exception
Click the Add a Device button in the Menu bar, and enter SSLD_000-00-01 at the Device Identification field and press Submit.

The following screen will appear, it shows that the device has been successfully added to the Platform.

**Opening Device Simulator to Add a Device**

In order to be able to use the SSLD-000-00-01 Device, the device needs to be simulated in the Device Simulator. To do this we have to create it. In the Firefox Browser, open the Device Simulator by going to the following URL:

https://localhost/web-device-simulator/devices

Click Add Device
Fill out the fields like this:

- **Device Identification**: SSLD_000-00-01
- **IP Address**: 127.0.0.1
- **Device Type**: SSLD
- **Protocol**: OSLP_ELSTER

Click Create Device

You should return to the Devices screen and see the message "Device with identification SSLD_000-00-01 was created."
Registering a Device

Now click on the newly created device and click the 'Register Device' button. After a while the message "Device identification with identification SSLD_000-00-01 was registered at XXXXXXXX" appears.

Then click the 'Confirm Registration' button. The message should read: "Device with identification SSLD_000-00-01 was confirmed to be registered."
Using 'SetLight' Switch the Light On

Now that the Device is known in the platform, and simulated in the Device-Simulator, the device can be used. Let's switch on the Light. Go to the Demo App and press the List button in the Menu. A list should appear, showing the device that has just been added using the Add Device button.

Click on the Manage button to access the Device Details.
Switch on the Light by setting the Light Value to 100 and by checking the 'LightOn' checkbox (as shown in the screenshot below).

Hit submit to submit the request to the Platform. The following screen should appear:
In the home screen of the OSLP device simulator, the lightbulb should light up for SSLD_000-00-01. This means that the request succeeded.

This step also concludes the installation manual.
Configuration

We have prepared a full set of configuration (properties files, certificates, key store, ...) which is available in our Config repository. This configuration can be used to setup a trial environment.
Add a device

Add a device

Platform uses single device calls.

In order to add a device to the platform, call the add device method in the device installation web service (for a specific domain) for each device.

Implementation may differ for different device types

Each device type may require its own set of attributes, security settings, etc. Each device type may have its own registration mechanism.

- To add a DLMS Device to the Platform, take a look at the documentation of the Soap call AddDevice as defined in the SmartMeteringInstallation.wsdl.
- To add an OSLP Device to the Platform, the Soap call defined in DeviceInstallation can be used, or the UpdateKey Request.

Please take a look at the chapter Testing the open smart grid platform in the installation manual for a detailed guide of how to add a OSLP device to the platform.

Additional Device actions

In the Domain Chapter of the documentation more information of the Web Service calls can be found for Adding devices, setting configuration, authorizations, schedules, firmware updates, etc.
Users

Soap Requests in GXF have fields for an User name and an Application name in the Header. These are only used for logging and auditing. There are no restrictions (e.g. Authorizations) with regard to these fields. However, it is recommended to use names for Logging and Auditing purposes.
Add a new organisation

Adding an Organisation to the platform

This chapter describes how to add a new Organisation to the platform. This includes creating a certificate for the new organisation.

Creating an Organisation

In SoapUI go to DeviceManagement under the Admin project. Click on request 1 under CreateOrganisation. Fill in the parameters like the example request below. Make sure to sign the request with the test-org.pfx and to use the test-org organisation in the request header.

This request creates a new organisation called "my-org":

```xml
  <ns:ApplicationName>APPLICATION_NAME</ns:ApplicationName>
  <ns:UserName>USER_NAME</ns:UserName>
  <ns:OrganisationIdentification>test-org</ns:OrganisationIdentification>
  <soapenv:Header>
    <ns:ApplicationName>APPLICATION_NAME</ns:ApplicationName>
    <ns:UserName>USER_NAME</ns:UserName>
    <ns:OrganisationIdentification>test-org</ns:OrganisationIdentification>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:CreateOrganisationRequest>
      <ns1:Organisation>
        <ns1:OrganisationIdentification>my-org</ns1:OrganisationIdentification>
        <ns1:Name>my-org</ns1:Name>
        <ns1:Prefix>MYO</ns1:Prefix>
        <ns1:FunctionGroup>ADMIN</ns1:FunctionGroup>
        <ns1:Enabled>true</ns1:Enabled>
        <ns1:Domains>COMMON</ns1:Domains>
        <ns1:Domains>PUBLIC_LIGHTING</ns1:Domains>
      </ns1:Organisation>
    </ns1:CreateOrganisationRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Authorise the new Organisation for the device

To use this new organisation to control the SSLD_000-00-01 device, the authorisations need to be updated. To do that the UpdateDeviceAuthorisations request will be used, it can be found under DeviceManagement in the admin project.

Fill in the parameters like shown below. The functionGroup will be set to AdHoc to authorise the 'my-org' organisation for the AdHoc functions.

Make sure to use the test-org as OrganisationIdentification in the request header, and to sign the request with the test-org.pfx.

```xml
  <ns:ApplicationName>APPLICATION_NAME</ns:ApplicationName>
  <ns:UserName>USER_NAME</ns:UserName>
  <ns:OrganisationIdentification>test-org</ns:OrganisationIdentification>
  <soapenv:Header>
    <ns:ApplicationName>APPLICATION_NAME</ns:ApplicationName>
    <ns:UserName>USER_NAME</ns:UserName>
    <ns:OrganisationIdentification>test-org</ns:OrganisationIdentification>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:UpdateDeviceAuthorisationsRequest>
      <ns1:DeviceAuthorisations>
        <ns1:DeviceIdentification>SSLD_000-00-01</ns1:DeviceIdentification>
        <ns1:OrganisationIdentification>my-org</ns1:OrganisationIdentification>
        <ns1:FunctionGroup>AD_HOC</ns1:FunctionGroup>
      </ns1:DeviceAuthorisations>
    </ns1:UpdateDeviceAuthorisationsRequest>
  </soapenv:Body>
</soapenv:Envelope>
```
Creating a certificate for the new organisation

Now that the 'my-org' organisation is authorised to use the SSLD_000-00-01 device, it is time to create a certificate for the my-org organisation. This certificate will be used to sign the requests.

Open a terminal and navigate to /home/dev/Sources/OSGP/Config/certificates/

A script has been created to create the certificates, execute it by running the following command in the terminal:

```
./create_client_cert.sh my-org 1234 1234
```

You should receive similar output as shown in the screenshot below.
Now that the certificate has been created, restart the tomcat server.

**Signing a request with the new certificate**

When the tomcat server is up and running again, go to SoapUi and add the new certificate to the public-lighting project: double-click on the project, go to the WS-Security Configurations tab and select the keystores tab. Click the '+' button and browse to the my-org.pfx certificate which can be found in `/home/dev/Sources/OSGP/Config/certificates/osgp-ca/certs/`
Now double-click on 'Request 1' in SetLight in PublicLightingAdHocManagement in the public-lighting project. Set the SSL Keystore to 'my-org.pfx' in the request properties so the request gets signed with the new certificate. Change the request parameters as shown in the example below:

```xml
  xmlns:ns1="http://www.opensmartgridplatform.org/schemas/publiclighting/adhocmanagement/2014/10"
>
  <soapenv:Header>
    <ns:ApplicationName>APPLICATION_NAME</ns:ApplicationName>
    <ns:UserName>USER_NAME</ns:UserName>
    <ns:OrganisationIdentification>my-org</ns:OrganisationIdentification>
  </soapenv:Header>
  <ns1:SetLightRequest>
    <!--type: Identification-->
    <ns1:DeviceIdentification>SSLD_000-00-01</ns1:DeviceIdentification>
    <!--1 to 6 repetitions:-->
    <ns1:LightValue>
      <!--Optional:-->
      <!--anonymous type-->
      <ns1:Index>0</ns1:Index>
      <!--type: boolean-->
      <ns1:On>true</ns1:On>
      <!--Optional:-->
      <!--anonymous type-->
      <ns1:DimValue>50</ns1:DimValue>
    </ns1:LightValue>
  </ns1:SetLightRequest>
</soapenv:Envelope>
```

Note the OrganisationIdentification is now set to 'my-org'. Send the new request, you should receive the following response:

```xml
  xmlns:ns2="http://www.opensmartgridplatform.org/schemas/pub"
>
  <SOAP-ENV:Header/>
  <ns2:SetLightAsyncResponse>
    <ns3:CorrelationUid>my-org|||SSLD_000-00-01|||20160805150420802</ns3:CorrelationUid>
    <ns3:DeviceId>SSLD_000-00-01</ns3:DeviceId>
  </ns2:SetLightAsyncResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Check the device-simulator to see if the dimValue of the SSLD_000-00-01 changed to 50.

You now have successfully created a new organisation, along with a certificate to sign the requests, and changed the device authorisations of the device to accept commands from the new organisation.
Web Services

Using the Open Smart Grid Platform Web Services

All the features of the open smart grid platform are accessible by its web services, as defined in the WSDL files. To use these services to communicate with devices through the platform, please keep in mind the following things.

- A Soap request will have to be generated from your Application.
- In the WSDL and xsd files of the Platform the Requests and its objects are defined.
- Fill in the parameters once you have an empty soap request for a certain function. The restrictions/requirements are defined in the WSDL files.
- The request should have a header where an ApplicationName, UserName and OrganisationId are defined. At the moment User and App name are not used in the Platform (Except for audability and logging). The OrganisationId, however, must match a known organisation within the Platform. This organisation must have the right authorities to make a certain request.
- Furthermore, the Request must be signed with a certificate from the Organization with the OrganisationId in the header.
- The requests use the Secure HTTP protocol (https).
- For Asynchronous requests, the Platform will respond with a correlationId after successfully making the request. Use this correlationId in the matching Response-Request to obtain the response from the device.

To learn more about the open smart grid platform's web services, please take a look at the Domain Documentation or for hands-on experience with the Platform's web services follow the UserGuide.

Rest

It is not possible to communicate with the Platform directly using REST web services. As mentioned above, the open smart grid platform uses SOAP (For reasons defined here). If you want to use REST for your front end, you can write an Integration Layer that serves as a Soap Client and exposes the Soap calls through Rest web services that you can access from your front end.

Flows

When using the SOAP Web service, there are 2 flows that can occur:

- some calls are synchronous: a response is returned immediately;
- other calls are a-synchronous: an initial response contains a correlation id, which can be used to obtain the actual response.
Deployment

Deployment options

Hosting the open smart grid platform in the cloud is possible, as well as on premises.

Active-active setup over multiple datacenters

If you like to setup an active-active installation over multiple datacenters. Make sure that the open smart grid platform database runs redundant over both datacenters (master-slave configuration).

Maintenance

There's not much maintenance that needs to be performed. Archiving some old log files, checking up on available disk space and creating a backup of the databases. Looking into the queues to see if there are no messages in the dead letter queue.
FAQ

How to start everything up? Make sure that PostgreSQL is running. Make sure that Apache HTTPD web server and Apache ActiveMQ are running. Then start Apache Tomcat application server as described in the installation manual.

Where are the log files? The Apache Tomcat application server logs can be found in /var/log/tomcat. The Apache HTTPD web server logs can be found in/var/log/httpd. The PostgreSQL log files can be found in /var/lib/pgsql/9.3/data/pg_log. The platform log files can be found in /var/log/osp/logs/.

Help: SELinux is preventing < something >? Make sure to set SELinux to 'permissive' mode. Then try again and it should work as SELinux will no longer enforce the current policy. Later, one can use the SELinux tools to create a proper policy that allows everything that was prevented before.

How to add or update a component? Make sure to place the properties file(s) for the component in /etc/osp. Add the locations of the properties file(s) to Apache Tomcat context.xml file. Add the war file to /usr/share/tomcat/webapps. Restart Apache Tomcat.

How to configure a component? Most (if not all) components of the open smart grid platform are de-coupled using queues. Configure the broker URL in the properties file and take notice of the queues that a component uses/needs. Make sure to double check the connectionstring for PostgreSQL.

How to configure URL's for a component? In this case the Apache HTTPD vhost needs to be updated. The vhost config file can be found in /etc/httpd/conf.d. We use redirects from HTTP to HTTPS and AJP proxy to send the requests to Apache Tomcat.

How to check up on Apache ActiveMQ? Apache ActiveMQ offers a web interface (the default port is 8161, default credentials: admin/admin). Using the web interface one can check the queues and especially the dead letter queue (DLQ).

How do I obtain a public key for a device? Public Keys are usually manufacture supplied for Smart Metering. For the Open Source part you can use the OSLP device simulator. In Sources/OSGP/Config/certificates/oslp/ you can find instructions for generating a OSLP device public key, and a folder with 5 pre-generated keys for test devices.

My code gives a lot of errors in Eclipse after importing Try the following things: run mvn clean install in the open-smart-grid-platform directory. Right-click on a project in Eclipse and select 'Maven -> Update project..'; select all projects and update.

The Vagrant script fails If you are receiving errors while downloading the ubuntu iso, sources, etc. or if the puppet script does not start; try running the Vagrant script again by typing vagrant destroy && vagrant up in the directory with the vagrant file.

I want to update my code from Github If you want to update your code, just run git pull in the repository you want to update. You can also create a fresh Virtual Machine using the vagrant installation, this procedure makes sure you have the most recent code.

Is a user required to consume platform services? No, an organization is required.

If your question is not in this list, please create an [issue on Github in the documentation repository](documentation repository)
Open Source Community

CHAPTER 3. Open Source and Community

We invite everyone to participate in the development of the open smart grid platform. There are multiple ways to support the project:

- submit issues about bugs or requested features
- solve issues
- develop new features
- write or improve the documentation

This chapter contains all the non-technical and general technical knowledge for developers to start contributing.
Start contributing

This is a guide to start contributing to the open smart grid platform project:

1. Make yourself comfortable with the open smart grid platform by e.g. installing it.
2. Read the documentation to get an idea of how the software works (architecture chapter), how the community works (this chapter).
3. Find an open issue to work on or fire an new issue.
4. Assign yourself to the issue and start contributing.
5. Start contributing by using the procedures mentioned in this chapter.
Developers 101

Developer tools, technical guidelines and continuous integration

Tools Used

The technology and tools used can be found in the Technology stack section.

Code Guidelines and Code Tests

- In Eclipse >Window >Preferences >Java >Code Style >Formatter, import the file /home/dev/Sources/OSGP/Config/code-format-settings/eclipse/code-format/JavaFormatter.xml
- In Eclipse >Window >Preferences >Java >Editor >Save actions, follow the images in /home/dev/Sources/OSGP/Config/code-format-settings/eclipse/save-actions.
- Follow GitFlow approach for branching
- Write behaviour driven tests using Cucumber and Gherkin, see the Integration-Tests
- In case you are not familiar with behaviour driven tests, include unit tests
- Fix SonarQube issues
- Issue pull request (preferable to development branch)

Continuous Integration

All changes pushed to GitHub are built by our build server. Pull requests to master branch or development branch are also built. SonarQube performs static code analysis to help improve the quality level of the code base.

- Jenkins buildserver: An open source continuous integration tool written in Java.
- SonarQube: SonarQube is an open platform to manage code quality.

Technical Conventions and Rules for New Code

This project is engineered, built and tested using Domain Driven Design and Behaviour Driven Tests.

- Use the Frameworks, don’t roll your own
- Single class, single responsibility
- Value objects are immutable
- Map generated objects to value objects or entities
- Transferring an object means using a DTO
- Use base classes for common logic
- Interfaces are good, but ‘impl’ is bad
- Extend classes, don’t expand classes
- Migrate databases using Flyway
- JMS for messaging
- Extend the authorization logic if needed, don’t bypass it
- Log errors/exceptions
- Add meaningful comments to the code
- Follow the code guidelines

Development Guide per Component

Web Service Adapters

The web service adapters use Spring Web Service, contract first. JAXB is used to generate Java classes from the XSD’s. All SOAP operations are bound to an endpoint. The incoming SOAP requests are authenticated by organization identification (plus certificates). Organization authorizations are checked for the desired operation. If the request is OK, a JMS message is sent to a domain adapter component.

Domain Adapters

Domain adapters contain business logic and persistence. Domain adapters process and forward the JMS message to the Core component.

Domain Components

Domain components contain entities and value objects.

Core

The Core component routes messages from domain adapter components to protocol adapter components and vice versa. Furthermore, it offers read-only database access for protocol adapter components.
Protocol Adapters

Protocol Adapter components translate a message from domain adapter components into a protocol message for a smart device. Protocol Adapter components send the protocol message to a smart device using a network connection. The response from the smart device is translated into a domain response message which will be sent to the Core components (which will route it to the domain adapter which issued the request).

OSLP

For the OSLP implementation, two components are used. The first component is the protocol adapter for the protocol. It can translate message into the protocol message for SSLP’s. Second there’s the signing-server component. It is responsible for signing the protocol message using the private key of the platform. The components communicate using a key-pair. The signing-server can handle multiple protocol adapter instances by utilizing a reply-to queue per protocol adapter instance. Since the protocol adapter component needs to be reachable from a network, it is a requirement that the private key may not be used by the protocol adapter directly. The signing server component can be deployed in such a way that no network access is available to this component, as the only coupling needed are the queues / the message broker.

Other Guides

Installation Guide

If a full installation is desired, have a look at our Installation Guide.

Create New Domain Guide

In order to add a new domain to OSGP, you can benefit from the information in the Create New Domain Guide.
Contributing to the code

Thank you for contributing to the Open Smart Grid Platform. Please keep the following in mind before submitting code.

Guidelines

Before code is merged it needs to comply with a number of guidelines:

1. Code should be as complete as possible (preferably no placeholders, TODO’s or FIXME’s)
2. Right formatting; code should follow the Coding Conventions (see 3.1.2)
3. Fixed/added unit tests where applicable
4. Javadocs added where applicable
5. Accepting pull-requests with SonarQube reports “Blocker” and “Critical” are not allowed
6. Comply with International open standards where possible (e.g. IEC standards)

Contributor License Agreement

We ask each of our contributors to sign our contributor license agreement (CLA). This has advantages for both parties, it gives you the assurance that your contribution will remain available under the Apache 2.0 license. Meanwhile, you give your code in license to us, so we can add your code to the open smart grid platform. And we know your contribution is entirely your work, so we don’t get in trouble with legal issues. Please read the CLA text carefully.

Submitting code

To submit changes to the open smart grid platform branches:

1. Create a fork of the open smart grid platform repo you will be working in
2. Make and commit your changes to your fork
3. Create a pull request to merge the changes into the right branch (see 4.1.4 for the branching strategy) If the changes fix a bug, mention the issue number in the commit message or pull request message (example: fixes 101, solved 87). If no ticket exists, create one beforehand. Afterwards, please update the relevant documentation in this GitBook.

Open Source Branching Strategy

The open smart grid platform's main branch is master. All major releases are tagged in this branch. Development is done in the development and feature branches. We use the GitFlow branching strategy. Find more information on this strategy here: GitFlow

The GitFlow workflow is somewhat complicated, but it has the advantage that it gives a clear overview of all previous releases and current development and thus helps to collaborate more efficiently. Please follow this strategy in your commits.

Pull requests: review process

Anyone can send in a pull request. Assign a maintainer or other developer with knowledge on this topic to accept/evaluate your pull request. You can view the SonarCube test results at (http://ci.opensmartgridplatform.org/sonarqube/) and the Jenkins continuous integration results at (http://ci.opensmartgridplatform.org)

If your code is a useful contribution and meets our quality standards (see section 3.1), it will be added to the open smart grid platform! Developers are in charge of judging this. Don't forget to update the documentation as well.
Contributing to documentation

Documentation Publication

This documentation is available in multiple formats.

Web

- Development branch publication
- Master branch publication
- Latest Pull-request publication

PDF

- Development branch publication in PDF

Contributing to documentation

The documentation is built using GitBook software from Markdown files in the documentation repository.

We encourage you to participate in improving the documentation. From corrected typos to new sections, every commit is appreciated. You can access the source files by clicking the "Fix this page"-button in the GitBook or by selecting the relevant Markdown-file in the documentation. You can commit your changes by sending a pull request.

1. Fork the repo, do work in a feature branch.
2. Issue a pull request.
3. Make sure the automated test suite succeeds. They will show-up in the pull request.
4. Sign the CLA using EasyCLA.
5. Assign a maintainer or other developer on this topic to accept/evaluate your pull request. The current maintainer can be found in the governance paragraph.

Some information on GitBook and using Markdown can be found here, more elaborate information on GitHub-flavored Markdown is found here. If you like to upload illustration, you can use git or Github.

If you are completely new to this and you need help to get started, file an issue in the documentation repository.

Chapters in the documentation

It may be obvious to you already from the index, however, here is an overview on what documentation goes in which chapter.

- Chapter 1 consist of an open smart grid platform introduction and architecture for potential users, architects and developers. The open smart grid platform website is used for basic product information.
- Chapter 2 contains the general userguide for open smart grid platform users
- Chapter 3 contains community related topics
- Chapter 4 contains information related to the open smart grid platform domains
- Chapter 5 contains information related to the protocols and simulators
- Chapter 6 shows the support options
- Chapter 7 code and documentation license

Versioning within the documentation

We have chosen to work with GitBook since it allows us to make different versions of documentation for each release. This is done by branching the files in the documentation repository. The master branch is used for releases only. The development branch is our main and current branch. If you like to improve the documentation, start a feature branch with your changes and send a pull request to the development branch.

Guidelines for new documentation

- The master branch is only used by major open smart grid platform releases
- Don't commit directly to the development branch, please do a pull request.
- We use the American spelling
- Please follow the used chapter and section numbering and apply it in your commits as well
- Currently we do not use image numbering, since it is too much of a hassle to keep it up-to-date. If you have a smart idea to do this, let us know!
- Give your (sub) document a relevant name or section with number
- Update SUMMARY.md if needed

Documentation inspiration
Inspiration on how to write good documentation can be found here: http://docs.writethedocs.org/
Communication and Contact

We choose Jira for most of our development communication. This keeps the communication central and topic central and connect to development effort. Please use Jira for issues and bugs.

If you want to get in touch to discuss non-technical subjects, send us an email to the LF energy GXF mailing list gxf@lists.lfenergy.org or open an issue on Github.

Jira was chosen due to its extensive features. The current Jira board is sponsored by Alliander for synergy reasons. Once we have multiple contributors outside Alliander we can move to a dedicated Jira instance or something else.

Once you get actively involved, you can request write permissions on the Jira board. You can request write access to the Jira issue board by sending a request to gxf@lists.lfenergy.org.

New Features

1. If there is a need (or wish!) for a new feature, add it as issue to Jira as a user story. Please provide a full description about the background of the problem. Split-up big user stories in multiple small user stories.

2. A developer can take on the issue and start working on it on voluntary base. If you need this feature and you have the money to pay for it, you can hire a developer and have the developer work on it. If open smart grid platform core components are involved, please discuss your change first with one of the developers/maintainers.

3. The developer makes a description on how he wants to fix the problem. Other developers can discuss the solution as well. If everybody agrees on the solution direction, the developer codes the solution and submits it (by sending in a pull request). The developer should also document the feature in the GitBook.

4. The maintainer can check the code (or assign this to someone else) and merge it with upstream releases.

Bug tracking

1. Find out as much as possible about the bug before reporting it. Please check on GitHub/Jira whether the bug has already been reported. Also, look for logs, error messages etc. Please include as much information as possible background on the bug and submit the bug on Github or Jira.

2. The maintainer makes sure that somebody will look at the bug, check for duplicates and thank the contributor for sending in the bug. If the bug turns out to be a duplicate, the issue will be closed.

3. A developer will try to reproduce the bug and will look for the root cause. The developer adds his findings to the issue. If the developer cannot reproduce the bug, the developer will contact the user for more information or/and login into the user's system (when possible for the user/developer). If it's impossible to re-produce the bug, the issue will be closed.

4. Otherwise, the developer will write a patch and tests the fix.

5. Once the patch is accepted (see Code review/test process), it will be shipped with the next release.

6. The maintainer than closes the issue.

Questions

If you have a question, please create an Github issue in the documentation repository or request access to Jira.

Report security issues

Due to the sensible nature of security issues e.g. zero days, we prefer a responsible disclosure. Security issues can be reported to gxf-owner@lists.lfenergy.org.
Governance

With the open smart grid platform we intend to have the right balance between a benevolent Dictator and a Formal Meritocracy in order to have a balanced decision-making process and to prevent unwanted dictators and everlasting discussions. The basic principle is that decisions are based on consensus. If this decision making process takes too long or a decision is required, the community council has the authority to make a decision. Grid eXchange Fabric complies with the LF Energy governance.

The governance and rules should be respected.

Community council / Technical Steering Committee

The community council consists of 5 members and has the authority to make decisions on all community related subjects.

The community council is responsible for:

- General ambitions, objectives and goals
- Principles and understandings
- Governance and consultation bodies
- Guidelines and procedures and tool selection
- Contribution (process) of individual members
- Architectural and (development) infrastructure choices
- Raise subjects/issuses that are important for the direction/development of the community

When the community grows, members of the community council can be elected. If the situation demands or requires it, the community council has the ability to establish sub councils for a specific subject, area of domain.

The community council consists of the following members:

- Robert Tusveld - Architect - Chairman
- Paul Houtman - Lead Architect
- Kevin Smeets - Maintainer
- Vacancy
- LF Energy TAC member - Vacancy

If you would like to join the community council, please contact us! Mention @OSGP/communitycouncil in an github issue. The (online) community council meetings will happen when needed.

Maintainers

Maintainers are responsible for maintaining parts of the code-base. Maintainers have the following responsibilities

- Coordinate development activity
- Make sure code/documentation reviews are being done
- Coordinate pull-requests
- Coordinate bug follow-ups
- Coordinate questions

In case of long discussions or arguments, maintainers or other can request a community council decision.

Current maintainers:

- Open smart grid platform and smart lighting domain: Kevin Smeets
- Smart metering domain: Sander van der Heijden
- Non-domain specific documentation & configuration: Ruud Lemmers
Code of Conduct

Like the technical community as a whole, the open smart grid platform community is made up of a mixture of professionals and volunteers from all over the world, working on every aspect of the mission - including mentor-ship, teaching, and connecting people.

Diversity is one of our huge strengths, but it can also lead to communication issues and unhappiness. To that end, we have a few ground rules that we ask people to adhere to. This code applies equally to founders, mentors and those seeking help and guidance.

This isn't an exhaustive list of things that you can't do. Rather, take it in the spirit in which it's intended - a guide to make it easier to enrich all of us and the technical communities in which we participate.

This code of conduct applies to all spaces managed by the GXF project (a LF energy project). This includes IRC, the mailing lists, the issue tracker, DSF events, and any other forums created by the project team which the community uses for communication. In addition, violations of this code outside these spaces may affect a person's ability to participate within them.

If you believe someone is violating the code of conduct, we ask that you report it by emailing the mailing list gxf@lists.lfenergy.org.

Be friendly and patient.

Be welcoming.

We strive to be a community that welcomes and supports people of all backgrounds and identities. This includes, but is not limited to members of any race, ethnicity, culture, national origin, color, immigration status, social and economic class, educational level, sex, sexual orientation, gender identity and expression, age, size, family status, political belief, religion, and mental and physical ability.

Be considerate.

Your work will be used by other people, and you in turn will depend on the work of others. Any decision you take will affect users and colleagues, and you should take those consequences into account when making decisions. Remember that we're a world-wide community, so you might not be communicating in someone else's primary language.

Be respectful.

Not all of us will agree all the time, but disagreement is no excuse for poor behavior and poor manners. We might all experience some frustration now and then, but we cannot allow that frustration to turn into a personal attack. It's important to remember that a community where people feel uncomfortable or threatened is not a productive one. Members of the open smart grid platform community should be respectful when dealing with other members as well as with people outside the open smart grid platform community.

Be careful in the words that you choose.

We are a community of professionals, and we conduct ourselves professionally. Be kind to others. Do not insult or put down other participants. Harassment and other exclusionary behavior aren't acceptable. This includes, but is not limited to:

- Violent threats or language directed against another person.
- Discriminatory jokes and language.
- Posting sexually explicit or violent material.
- Posting (or threatening to post) other people's personally identifying information (“doxing”).
- Personal insults, especially those using racist or sexist terms.
- Unwelcome sexual attention.
- Advocating for, or encouraging, any of the above behavior.
- Repeated harassment of others. In general, if someone asks you to stop, then stop.

When we disagree, try to understand why.

Disagreements, both social and technical, happen when people get passionate about what they are doing. It is important that we resolve disagreements and differing views constructively. Remember that we’re different. The strength of open smart grid platform comes from its varied community, people from a wide range of backgrounds. Different people have different perspectives on issues. Being unable to understand why someone holds a viewpoint doesn’t mean that they’re wrong. Don’t forget that it is human to err and blaming each other doesn’t get us anywhere. Instead, focus on helping to resolve issues and learning from mistakes.

This Code of Conduct is based on the Django Code of Conduct
Foundation

Foundation

Grid eXchange Fabric (GXF) is a project of LF Energy. LF Energy is part of the Linux foundation.
Domains

Chapter 4 Domain This chapter describes the separate domain in the open smart grid platform.

Web Service Adapters

The web service adapters use Spring Web Service, contract first. JAXB is used to generate Java classes from the XSD's. All SOAP operations are bound to an endpoint. The incoming SOAP requests are authenticated by organization identification (plus certificates). Organization authorizations are checked for the desired operation. If the request is OK, a JMS message is sent to a domain adapter component.

Domain Adapters

Domain adapters contain business logic and persistence. Domain adapters process and forward the JMS message to the Core component.

Domain Components

Domain components contain entities and value objects.
Admin

The open smart grid platform core and admin functions for device management.

Scope

This core and admin domain contains all generic web services that can be used in other domains. Most generic services relate to device management including powerful device authorization management.

Common webservices

<table>
<thead>
<tr>
<th>Operation</th>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeviceInstallation.wsdl</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AddDevice</td>
<td>DeviceIdentification</td>
<td>-</td>
</tr>
<tr>
<td>FindDevicesWhichHaveNoOwner</td>
<td>-</td>
<td>Devices</td>
</tr>
<tr>
<td>FindRecentDevices</td>
<td>-</td>
<td>Devices</td>
</tr>
<tr>
<td>StartDeviceTest</td>
<td>DeviceIdentification</td>
<td>-</td>
</tr>
<tr>
<td>StopDeviceTest</td>
<td>DeviceIdentification</td>
<td>-</td>
</tr>
<tr>
<td>UpdateDevice</td>
<td>DeviceIdentification</td>
<td>-</td>
</tr>
<tr>
<td>GetStatus</td>
<td>DeviceIdentification</td>
<td>Status</td>
</tr>
<tr>
<td>DeviceManagement.wsdl</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ChangeOrganisation</td>
<td>Organisation</td>
<td>-</td>
</tr>
<tr>
<td>CreateOrganisation</td>
<td>Organisation</td>
<td>-</td>
</tr>
<tr>
<td>FindAllOrganisations</td>
<td>-</td>
<td>Organisations</td>
</tr>
<tr>
<td>FindDeviceAuthorisations</td>
<td>DeviceIdentification</td>
<td>DeviceAuthorisations</td>
</tr>
<tr>
<td>FindDevices</td>
<td>PageSize, Page</td>
<td>Devices, Page</td>
</tr>
<tr>
<td>FindEvents</td>
<td>DeviceIdentification, From, Until, PageSize, Page</td>
<td>Events, Page</td>
</tr>
<tr>
<td>FindMessageLogs</td>
<td>DeviceIdentification, Page</td>
<td>MessageLogPage</td>
</tr>
<tr>
<td>RemoveDevice</td>
<td>DeviceIdentification</td>
<td>-</td>
</tr>
<tr>
<td>RemoveOrganisation</td>
<td>Organisation</td>
<td></td>
</tr>
<tr>
<td>SetEventNotifications</td>
<td>DeviceIdentification, EventNotifications</td>
<td>-</td>
</tr>
<tr>
<td>SetOwner</td>
<td>DeviceIdentification, OrganisationIdentification</td>
<td>-</td>
</tr>
<tr>
<td>UpdateDeviceAuthorisations</td>
<td>DeviceAuthorisations</td>
<td></td>
</tr>
<tr>
<td>ActivateOrganisation</td>
<td>Organisation</td>
<td>-</td>
</tr>
<tr>
<td>SetDeviceLifecycleStatus</td>
<td>DeviceIdentification, DeviceLifecycleStatus</td>
<td>-</td>
</tr>
<tr>
<td>FirmwareManagement.wsdl</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GetFirmwareVersion</td>
<td>DeviceIdentification</td>
<td>FirmwareVersion</td>
</tr>
<tr>
<td>UpdateFirmware</td>
<td>DeviceIdentification, FirmwareIdentification</td>
<td>-</td>
</tr>
<tr>
<td>ConfigurationManagement.wsdl</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GetConfiguration</td>
<td>DeviceIdentification</td>
<td>Configuration</td>
</tr>
<tr>
<td>SetConfiguration</td>
<td>DeviceIdentification, Configuration</td>
<td>-</td>
</tr>
<tr>
<td>ScheduleManagement.wsdl</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetSchedule</td>
<td>DeviceIdentification, Schedules, Page</td>
<td>-</td>
</tr>
<tr>
<td>SetTariffSchedule</td>
<td>DeviceIdentification, Schedules, Page</td>
<td>-</td>
</tr>
<tr>
<td>AdHocManagement.wsdl</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetReboot</td>
<td>DeviceIdentification</td>
<td>-</td>
</tr>
</tbody>
</table>

- Core WSDL's
- Core XSD schema's

Admin webservices

- Admin WSDL's
- Admin XSD schema's
Smart lighting

This domain covers the use of the open smart grid platform for smart lighting.

Scope

The goal of this domain is to control, monitor and manage (street) lights at scale. It allows streetlight owners to control/manage the (street) lights in an more intelligent way compared to ripple control technology.

Features

This domain consist of Switching schedules, groups, light sensors, manual switching and monitoring of a typical public lighting use-case.

PublicLighting webservice

<table>
<thead>
<tr>
<th>Operation</th>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeviceMonitoring.wsdl</td>
<td>DeviceIdentification</td>
<td>PowerUsageData</td>
</tr>
<tr>
<td></td>
<td>DeviceIdentification, TimePeriod, DeviceID</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HistoryTermType, Page</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GetActualPowerUsage</td>
<td>DeviceIdentification</td>
<td>PowerUsageData, PageInfo</td>
</tr>
<tr>
<td></td>
<td>DeviceIdentification, TimePeriod, DeviceID</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HistoryTermType, Page</td>
<td></td>
</tr>
<tr>
<td>GetPowerUsageHistory</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PublicLighting AdHocManagement</td>
<td>DeviceIdentification, TransitionType, Time</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>SetTransition</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FindAllDevices</td>
<td>DeviceIdentification</td>
<td>DevicePage</td>
</tr>
<tr>
<td></td>
<td>TransitionType, Time</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>GetStatus</td>
<td>DeviceIdentification</td>
<td>LightValues, PreferredLinkType,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ActualLinkType, LightType, EventNotifications</td>
</tr>
<tr>
<td>ResumeSchedule</td>
<td>DeviceIdentification, Index,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Immediate</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetLight</td>
<td>DeviceIdentification, LightValue</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PublicLighting Schedule Management</td>
<td>DeviceIdentification, Schedules, Page</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- [PublicLighting WSDL's](#)
- [PublicLighting XSD schema's](#)
Use cases

Example use-case for this domain

Up-to-date information on use-cases can be found on the Grid eXchange Fabric website.

Reference implementation in The Netherlands: Flexible system for operating public lighting (FlexOVL)

FlexOVL, a new and flexible switching system of public lighting delivers more control for municipalities and is the first solution which is powered by the Open Smart Grid Platform.

Technical drivers for Alliander

- Replacing Ripple Control
- Decrease future investments
- Decrease outages

Customer drivers (Municipalities)

- Be more in control, by controlling switching times themselves
- Resolve power failures faster, through up-to-date information
- Reduction in costs, through energy saving and more efficient maintenance and management
- No vendor lock-in, not dependent on one supplier

Implementation/roll-out

- Small scale roll-out started January 2015
- 200 Sub Stations will be fitted with an SSLD to control public lighting and tariff switching
- 15 municipalities in the Liander grid operator area will be participating
- Goal is to allow municipalities to use the application, give feedback and to see if the services offered to municipalities are adequate

- Large scale roll-out will start around 2016
- The entire Liander grid operator area will use SSLD's to control all public lighting and tariff switching
- About 25,000 Sub Stations (middenspanningsruimtes)
- About 800,000 street lights will be switched by the SSLD's mounted in the 25,000 Sub Stations

FlexOVL web application (not open source available)

Municipalities are free to choose their own (web)application (using the web services of the Open Smart Grid Platform), or they could use the default web application developed by Alliander.
**Functionality of the default web application as used by grid operator Liander (example)**

- Create switching schedules and assign those schedules to one or more SSLD's
- Create groups of SSLD's in order to be able to assign schedules to many SSLD's at once
- On demand switching of public lighting
- Review current status of an SSLD in order to review public lighting and tariff switching states
- Abilities to monitor power consumption of public lighting (available if the SSLD is fitted with an Electricity Meter)
- Monthly report offering insight into switch moments and power consumption
Light Schedules

Light Schedules

Schedules for light switching can be set using Set Schedule requests from the Public Lighting Schedule Management web service. For brevity the XML element and type names in the descriptions below will not include the namespace (which will typically be "http://www.opensmartgridplatform.org/schemas/publiclighting/schedulemanagement/2014/10").

A switching schedule is defined by a number of declarations of switching moments (also known as schedule entries). The SetScheduleRequest defines the schedule, where Schedules of type Schedule define the entries. A complete schedule for a device as set with the Set Schedule request can have 1 up to 50 entries. Each schedule entry defines a moment on a day when certain relays on a device are switched on or off. Whether or not a switch action defined in a schedule entry is executed may not only depend on the entry itself. Other switch moments from the schedule that are close in time compared to an entry may cause switching to be skipped.

A more detailed description of the components defining a schedule entry is in the sections below:

- **week day**
- **time**
  - explicitly configured fixed time
  - time of sunrise or sunset
  - time of sunrise or sunset combined with sensor input
- **light value**

**Week Day**

The value of WeekDay is used to indicate on which days the schedule entry may trigger switch actions.

<table>
<thead>
<tr>
<th>WeekDay</th>
<th>May trigger a switch action on</th>
</tr>
</thead>
<tbody>
<tr>
<td>MONDAY</td>
<td>Mondays</td>
</tr>
<tr>
<td>TUESDAY</td>
<td>Tuesdays</td>
</tr>
<tr>
<td>WEDNESDAY</td>
<td>Wednesdays</td>
</tr>
<tr>
<td>THURSDAY</td>
<td>Thursdays</td>
</tr>
<tr>
<td>FRIDAY</td>
<td>Fridays</td>
</tr>
<tr>
<td>SATURDAY</td>
<td>Saturdays</td>
</tr>
<tr>
<td>SUNDAY</td>
<td>Sundays</td>
</tr>
<tr>
<td>WEEKDAY</td>
<td>weekdays (Monday to Friday)</td>
</tr>
<tr>
<td>WEEKEND</td>
<td>weekend days (Saturday or Sunday)</td>
</tr>
<tr>
<td>ABSOLUTEDAY</td>
<td>the day specified in startDay</td>
</tr>
<tr>
<td>ALL</td>
<td>any day</td>
</tr>
</tbody>
</table>

**Time**

Each schedule entry can cause switching at a single time during the day. There are a number of ways in which this time can be specified, starting with ActionTime.

<table>
<thead>
<tr>
<th>ActionTime</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSOLUTETIME</td>
<td>a fixed time is set for the switching moment in Time</td>
</tr>
<tr>
<td>SUNRISE</td>
<td>switching at sunrise at the location of the device</td>
</tr>
<tr>
<td>SUNSET</td>
<td>switching at sunset at the location of the device</td>
</tr>
</tbody>
</table>

For ActionTime values SUNRISE or SUNSET the value of TriggerType specifies what the actual switching time should be.

<table>
<thead>
<tr>
<th>TriggerType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIGHT_TRIGGER</td>
<td>astronomical time with sensor input determines the actual switching time</td>
</tr>
<tr>
<td>ASTRONOMICAL</td>
<td>the calculated astronomical time for sunrise or sunset is the switching time</td>
</tr>
</tbody>
</table>

**Fixed Time**

For ActionTime ABSOLUTETIME a fixed time can be set for the switching moment as Time. The Time value needs to be formatted in a way the protocol implementations can handle. For the currently listed
implementations, you should be fine when you use a format from:

- \texttt{hh:mm};
- \texttt{hh:mm:ss};
- \texttt{hh:mm:ss.SSS}

With \texttt{hh} from 00 to 23, \texttt{mm} from 00 to 59, \texttt{ss} from 00 to 59 and \texttt{SSS} from 000 to 999.

Some protocols may accept more precise time formats than they support. The IEC61850 implementation for instance, will silently apply only the hours and minutes from any of the formats listed above.

**Astronomical Time**

For \texttt{ActionTime SUNRISE} or \texttt{SUNSET} with \texttt{TriggerType ASTRONOMICAL} the astronomical sunrise or sunset time (as calculated by the switching device, based on its longitude and latitude) will be used to determine the switching moment.

**Astronomical Offsets**

If an astronomical offset is configured, it has to be added to the calculated astronomical time to determine the time to be used as the switching moment.

For positive offset values, the astronomical time for the switching moment will be the configured amount of minutes after the calculated astronomical sunrise or sunset time, while for negative values the astronomical time used will be the number of minutes before the calculated astronomical sunrise or sunset time.

**Astronomical Sunrise Offset**

The astronomical sunrise offset is applied with entries with \texttt{ActionTime SUNRISE} and \texttt{TriggerType ASTRONOMICAL}.

The following picture is an example of switching off at 07:30; the calculated astronomical sunrise (say at 07:15 for the day shown) plus 15 minutes (configured as \texttt{AstronomicalSunriseOffset 15}).

![Astronomical Sunrise Offset Example](image)

**Astronomical Sunset Offset**

The astronomical sunset offset is similar to the astronomical sunrise offset, except that it is applied with entries with \texttt{ActionTime SUNSET} and \texttt{TriggerType ASTRONOMICAL}.

**Astronomical Time With Sensor**

For \texttt{ActionTime SUNRISE} or \texttt{SUNSET} with \texttt{TriggerType LIGHT_TRIGGER} the calculated astronomical sunrise or sunset time will be used as a reference time with a trigger window to determine the switching moment.

The astronomical time itself is calculated in the same way as with astronomical time (without light sensor input).

Switching happens within a configured trigger window around the astronomical time, at a moment that is influenced by a signal from a light sensor.

**Trigger Window**

The \texttt{TriggerWindow} with \texttt{minutesBefore} and \texttt{minutesAfter} defines a window of time around an astronomical sunrise or sunset time with sensor.

Switching will occur at the start of the window when light sensor input is received before the window.

Switching will occur at the end of the window when light sensor input is not received before the end of the window.

Switching will occur at the time light sensor input is received, when this input is received within the window.

Light sensor input in the conditions above means the sensor trigger for \texttt{light} when the schedule entry is switching off, and the sensor trigger for \texttt{dark} when switching on.

**Astronomical Time With Sensor Signal Within The Trigger Window**

Light Schedules
The following picture is an example where the light sensor reports dark within the trigger window for a schedule entry for astronomical time with sensor signal. Switching on occurs at the time the sensor input is received. Note that for this example this could have been at any time between 19:45 and 20:30 (15 minutes before to 30 minutes after the astronomical sunset, calculated to occur at 20:00 on the day shown).

Astronomical Time With Sensor Signal Before The Trigger Window Opens

The following picture is an example where the device has received a light sensor report before the trigger window opens for a schedule entry for astronomical time with sensor signal. Switching off occurs at the start of the trigger window.

Astronomical Time Without Sensor Signal Before The Trigger Window Closes

The following picture is an example where the device has not received a light sensor report before the trigger window closes for a schedule entry for astronomical time with sensor signal. Switching on occurs at the end of the trigger window.
Minimal Burning Time

For certain types of lighting it may be undesirable to switch the lights on only for a short period of time, after which they are switched off again. In such a case the action of switching the lights on will be suppressed if `minimumLightsOn` is set with a positive number of seconds, and the action switching the lights off again is expected within this time period.

The minimal burning time is always regarded with respect to an **expected time** for a switching moment that switches a relay on in comparison with the **actual time** for a switching moment that the same relay will be switched off again. Switching on will be skipped if switching off is expected to occur within a number of minutes set as `minimumLightsOn` with the schedule entry that switches the relay on.

**Minimal Burning Time With Astronomical Offset**

This example shows the minimal burning time preventing the *morning lights* to be switched on at a fixed time because switching off at the calculated time of astronomical sunrise (with offset) would happen before passing of the minimum number of minutes the lights should be kept on.

**Minimal Burning Time With Light Sensor Trigger Window**

This example shows the minimal burning time preventing the *morning lights* to be switched on at a fixed time because switching off at the start of the trigger window around the calculated time of astronomical sunrise would happen before passing of the minimum number of minutes the lights should be kept on.

**Light Value**

Each schedule entry may include 1 to 6 `LightValue` elements. These light values determine the relay to switch, whether the relay should be switched on or off, and whether the lights with a relay should be dimmed (and by how much).

- **Index**: 0 for all light switching relays in the device, or 1 to 6 for numbered relays (the index should indicate an existing relay that is used for light switching).
• **On**: true if this entry is for switching on the relay(s) identified by Index; false for switching off.
• **DimValue**: optional percentage set as number 1 to 100 indicating a dim value; will be ignored when the protocol or switching device does not support dimming.

### Common Light Scheduling Patterns

Here are some examples of patterns that are common with light schedules. The patterns are formed by combinations of schedule entries that switch on or off lights controlled by a certain relay on the switching device.

#### All Night Lights

All night lights is a name for lights that are turned on around sunset and keep burning all night until they are switched off again around sunrise.

The all night lights are switched by a pair of schedule entries:

- one entry switching on at **astronomical sunset time** with optional offset or **light sensor trigger**.
- another entry switching off again based on **astronomical sunrise time** with optional offset or **light sensor trigger**.

#### Morning Lights

Morning lights is a name used for lights that are switched on a short period in the morning hours of a day to illuminate a period before or around the morning twilight.

The morning lights are switched by a pair of schedule entries:

- one entry switching on at **fixed time**;
- another entry switching off again based on **astronomical sunrise time** with optional offset or **light sensor trigger**.

#### Fixed Time And Sunrise Interaction

Depending on the location of the device the time of sunrise may vary quite a bit throughout the year. Because of this it is possible that what for some period would be a very reasonable schedule, is a questionable schedule (possibly to be considered invalid) in another season.

During the summer in the Netherlands for example sunrise can be as early as approximately 05:15, while during the winter the sun may rise even a little later than 08:45.

For this example we will assume configuration for the morning lights to switch on at fixed time of say 06:00. This is a time after the earliest sunrise in the year, but well before the latest sunrise in the year.

To complete the morning lights configuration, a second switching moment is configured to switch the lights off at sunrise. With this setup the lights will be switched off after having been on for almost three hours at some time in the winter (for instance from 06:00 to 08:45).

During summer at some days the lights will not be switched off in the morning at all because they were turned on (at 06:00) after sunrise (switching off at any time before 06:00, for instance at 05:30).
Whether the lights stay on all day in the summer or not is something to be looked into. A switching device may have logic to deal with this situation figuring out the switch off belongs with the later switching moment to turn the lights on, and decide not to switch on. If not, some validation may be needed to enforce such schedules not to be configured.

**Evening Lights**

Evening lights is a name used for lights that are switched on a short period in the evening hours of a day to illuminate a period after or around dusk. This is similar to the [morning lights](#), but in the evening instead of the morning, and the fixed time moment comes (normally) after the switch action around sunset.

The evening lights are switched by a pair of schedule entries:

- one entry switching on at *astronomical sunset time* with *optional offset* or *light sensor trigger*.
- another entry switching off again based on *fixed time*.

**Fixed Time And Sunset Interaction**

Depending on the location of the device the time of sunset may vary quite a bit throughout the year. Because of this it is possible that what for some period would be a very reasonable schedule, is a questionable schedule (possibly to be considered invalid) in another season. See the [explanation around sunrise](#) for a graphical example.

**Evening/Morning Lights**

A combination of morning lights and evening lights can be configured for a relay if the lights may be turned off for a period in the late night and early morning, as opposed to the all night lights that keep on burning all through the night.
Validity

The GXF Public Lighting Schedule Management web service does not do much validation, other than checking authorizations for the device identified by the DeviceIdentification from the SetScheduleRequest and whether the request conforms to its XML schema definitions.

If for certain applications more constraints are desirable, it is left up to those applications to make sure the requests made to the platform conform to those additional constraints. Examples of such constraints, that are not enforced by GXF, could be:

- no duplicated schedule entries;
- no schedule entries canceling the switch actions of other entries within some time window;
- schedule entries may be required for all days of the week;
- switching on and off might be required to happen each day in equal number of times and alternately;
- checking expected actions around daylight saving change;
- checking expected switching actions for days with the longest or shortest number of hours of daylight;
- light value indexes map to existing light relays on the device the schedule is set on;
- constraints from applying the provided input with specific devices or protocols.

Protocol Implementations For Light Schedules

- IEC61850
- OSLP v0.6.1
Tariff switching

Tariff switching

This covers the services for tariff switching domain using the open smart grid platform.

Scope

This domain allows tariff switching. It allows a relay to switch when a tariff changes. This domain could be used to replace ripple control tariff signals.

Webservices

<table>
<thead>
<tr>
<th>Operation</th>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>TariffSwitchingAdHocManagement</td>
<td>GetDevices</td>
<td>GetStatus</td>
</tr>
<tr>
<td>GetStatus</td>
<td>DeviceIdentification</td>
<td>Status</td>
</tr>
<tr>
<td>TariffSwitchingScheduleManagement</td>
<td>SetSchedule</td>
<td>DeviceIdentification, Schedules, Page</td>
</tr>
</tbody>
</table>

- Tariff switching WSDL's
- Tariff switching XSD schema's

TARIFF (normal) vs. TARIFF_REVERSED

When configuring a device via the platform to switch relays according to a tariff schedule, the device can be instructed to switch the tariff relay normally ("TARIFF") or reversed ("TARIFF_REVERSED").

The devices themselves are unaware of the difference between TARIFF and TARIFF_REVERSED. When sending a setScheduleRequest message for a tariff schedule to the platform, the tariff switching domain adapter checks if the device relay(s) are configured with TARIFF_REVERSED. If so, the tariff switching domain adapter will invert the relay value for all tariff schedule entries before the tariff schedule is sent to the device.

When two devices have the same schedule, while one device is using TARIFF and the other TARIFF_REVERSED, the state of their tariff relays will always be the opposite of each other.

Communicated/Stored values

The values as shown in the table below will be returned, when getting the status from a device or from the platform.

<table>
<thead>
<tr>
<th>Type</th>
<th>State</th>
<th>Relay powered</th>
<th>Returned by Device</th>
<th>Returned by Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>TARIFF</td>
<td>LOW</td>
<td>yes</td>
<td>on = true</td>
<td>High = false</td>
</tr>
<tr>
<td></td>
<td>HIGH</td>
<td>no</td>
<td>on = false</td>
<td>High = true</td>
</tr>
<tr>
<td>TARIFF_REVERSED</td>
<td>LOW</td>
<td>no</td>
<td>on = false</td>
<td>High = false</td>
</tr>
<tr>
<td></td>
<td>HIGH</td>
<td>yes</td>
<td>on = true</td>
<td>High = true</td>
</tr>
</tbody>
</table>
Microgrids

Microgrids documentation

The Open Smart Grid Platform act as an central component to monitor and control microgrids.

Scope

The goal of this domain is to control and monitor microgrids.

Features

Currently, the following features are available within the Open Smart Grid Platform:

Get Data

Get data is used to retrieve measurement and profile data from the device

Set Data

Set data is used to set setpoints and profiles on the device

Notifications

When either report data or the result for a request is available, a notification is sent to a client, after which the client will be able to obtain the data or result by sending an 'async' message.

Reporting

When a device is connected it will periodically push measurement reports (and send trigger-based status reports) to OSGP. OSGP will inform the client via a notification, after which the data can be retrieved in a way similar to GetData (using the GetDataAsync message). In order to determine whether all report data are received, the response of a GetDataAsync message will (in case of a report) contain report metadata consisting of a report id, sequence number and time of entry. (XSD is already updated with report metadata, returning the report metadata from OSGP is not yet implemented)

Messages

- **GetData** is a request to retrieve measurement and profile data from a device.
- **GetDataAsync** is a request to retrieve the result of the GetData request or to retrieve report data pushed by the device.
- **SetData** is a request to set setpoints and profiles on a device.
- **SetDataAsync** is a request to retrieve the result of the SetData request.

WSDLs

- WSDL’s and schema’s

Cucumber test

Functionality like **GetData** can now be tested, with the Cucumber framework, using the protocol-simulator-iec61850. This simulator can be started from the Cucumber tests, and is configured with its own properties file.

Multiple Server names

By default the RTU device is configured with the servername: **WAGO61850Server**. This name also appears in the icd file, that is used by the RTU device. The name of this icd file, is configured in a properties file. Multiple server names are now supported, with the introduction of the new column: server_name in the iec61850_device table. If this value is null, the default servername (WAGO61850Server) is used, otherwise the servername from the database is used (eg: 'WAGO123'). In that case another corresponding icd file, in which this servername is used, must be configured!
Distribution automation

Distribution Automation

The Open Smart Grid Platform can also be used in the monitoring of a variety of components in substations; RTUs, switches, transformers, etc. Often, an RTU or Remote Terminal Unit is used as a central information hub in a substation. The RTU is connected to one or more sensors or devices that can measure any kind of information. Usually, the focus is on measuring power quality values, temperature and other 'health' variables, but any kind of measurable data can be read through OSGP.

Scope

The goal of this domain is to control and monitor substations; the current focus is on health status information and power quality data, but this may be extended in the future.

Features

Currently, the following features are available within the Open Smart Grid Platform using the IEC 61850 protocol;

Get PQ Values

Get PQ Values retrieves the actual, current PQ values from a device. Examples of PQ values are Current, Voltage, Reactive Power, Active Power, etc. These examples merely serve as an indication of what is possible; OSGP does not impose any restriction on the number or content of variables that can be read. The outline of what should be measured is configured on the device and in the application that reads the data.

Get Health Status

Retrieves the current health status of a device. This is useful in a monitoring application.

Get Device Model

Retrieves the device model or metadata of a device. This includes the variables that can be measured, the information structure of the device, etc.

Notifications

When either report data or the result for a request is available, a notification is sent to a client, after which the client will be able to obtain the data or result by sending an 'async' message. A notification message always contains the correlation ID of the original request; the client can retrieve the result using this correlation ID.

Messages

- GetPQValues is a request to retrieve PQ values from a device.
- GetPQValuesAsync is a request to retrieve the result of the GetPQValues request or to retrieve report data pushed by the device.
- GetHealthStatus is a request to retrieve the health status of a device.
- GetHealthStatusAsync is a request to retrieve the result of a GetHealthStatus request.
- GetDeviceModel is a request to retrieve the device model from a device
- GetDeviceModelAsync is a request to retrieve the result of a GetDeviceModel request.

WSDLs

- WSDL's and schema's
SmartMetering

SmartMetering Documentation (Beta version)

This chapter describes the SmartMetering domain including the web services. Currently the web services of the beta version are described, since the web services have not yet officially been released. Information on the DLMS device simulator can be found in the DLMS protocol section.

Scope

The goal of this domain is to read and manage millions of smart meters. This includes smart meter installation, firmware updates, smart meter removal, read smart meter values, time synchronisation, etc. Everything that is needed to professionally manage millions of smart meters is or can be included in this domain.

Features

Currently, the following Smart Metering features are available within the open smart grid platform:

- Add smart meter to the platform, so the device is known and additional actions can be performed for the device
- Process shipment file, which adds several smart meters to the platform along with all needed information
- Synchronize time between smart meters and head-end system, in case the smart meter adjusts its time, some events will be logged
- Retrieve events from the smart meter, several event logs are available
- Retrieve periodic meter reads from the smart meter

Generic functionality

- `bypass retry` operations can be given the flag 'bypass retry'. Which means that an operation will not be retried in case of an error.
- `priority` operations can be given a priority from 0 to 9, default is 4. Higher values causes messages to be processed faster.
- `scheduling` operations can be scheduled for a certain time.
- `bundling` operations can be combined in a Bundle.

Messages

SmartMetering AdHoc

- `SynchronizeTime` is an operation to synchronize the date and time on a device. The date and time are retrieved from the server and sent to the device.
- `GetSynchronizeTimeResponse` is an operation which returns the response from the `SynchronizeTime` operation.
- `RetrieveAllAttributeValues` is an operation to obtain all the attributes of the whole tree of objects from an E-meter.
- `GetRetrieveAllAttributeValuesResponse` is an operation which returns the response from the `RetrieveAllAttributeValues` operation.
- `GetSpecificAttributeValue` is an operation to obtain a specific attribute value from an ObisCode from an E-meter.
- `GetSpecificAttributeValueResponse` is an operation which returns the response from the `GetSpecificAttributeValue` operation.
- `GetAssociationLnObjects` is an operation to get the associated ln objects.
- `GetGetAssociationLnObjectsResponse` is an operation which gets the response from the `GetAssociationLnObjects` operation.
- `ScanMbusChannels` is an operation to get the M-Bus Short ID attribute values for all four channels from an E-meter.
- `ScanMbusChannelsResponse` is an operation which returns the response from the `ScanMbusChannels` operation.

SmartMetering Configuration

- `SetSpecialDays` is an operation to set a dayId profile and its tariffs for a specific date on a device.
- `GetSetSpecialDaysResponse` is an operation which returns the response from the `SetSpecialDays` operation.
- `SetConfigurationObject` is an operation to set ConfigurationObject settings on a device to specify behaviour and connection options.
- `GetSetConfigurationObjectResponse` is an operation which returns the response from the `SetConfigurationObject` operation.
- `GetConfigurationObject` is an operation to retrieve a ConfigurationObject from a device.
- `GetGetConfigurationObjectResponse` is an operation which returns the response, a ConfigurationObject, from the `GetConfigurationObject` operation.
- `SetPushSetupAlarm` is an operation that pushes received alarm messages to OSGP.
- `GetSetPushSetupAlarmResponse` is an operation which returns the response from the `SetPushSetupAlarm` operation.
- `SetPushSetupSms` is an operation to set an endpoint in a device which tells the device where to connect to when it is waked up.
- `GetSetPushSetupSmsResponse` is an operation which returns the response from the `SetPushSetupSms` operation.
SetAlarmNotifications is an operation to set the types of alarm notifications that must be notified from the device when they occur.

GetSetAlarmNotificationsResponse is an operation which returns the response from the SetAlarmNotifications operation.

SetEncryptionKeyExchangeOnGMeter is an operation to transfer and set a G-meter key on a device.

GetSetEncryptionKeyExchangeOnGMeterResponse is an operation which returns the response from the SetEncryptionKeyExchangeOnGMeter operation.

SetMbusUserKeyByChannel is an operation to set the M-Bus encryption key on an M-Bus device by using the E-meter device identification and channel from the G-meter.

GetMbusUserKeyByChannelResponse is an operation which returns the response from the SetMbusUserKeyByChannel operation.

GetMbusEncryptionKeyStatus is an operation to retrieve the encryption key status for a M-Bus device.

GetMbusEncryptionKeyStatusResponse is an operation which returns the response from the GetMbusEncryptionKeyStatus operation.

GetMbusEncryptionKeyStatusByChannel is an operation to get the M-Bus encryption key status from an M-Bus device by using the E-meter device identification and channel from the G-meter.

GetMbusEncryptionKeyStatusByChannelResponse is an operation which returns the response from the GetMbusEncryptionKeyStatusByChannel operation.

GetActivityCalendar is an operation to set several parameters on an E-meter such as tariffs per day in a week profile.

GetSetActivityCalendarResponse is an operation which returns the response from the GetActivityCalendar operation.

GetAdministrativeStatus is an operation to retrieve the current AdministrativeStatus setting.

GetGetAdministrativeStatusResponse is an operation which returns the response from the GetAdministrativeStatus operation.

SetAdministrativeStatus is an operation to set the AdministrativeStatus.

GetSetAdministrativeStatusResponse is an operation which returns the response from the SetAdministrativeStatus operation.

GetFirmwareVersion is an operation to retrieve the firmware version(s).

GetGetFirmwareVersionResponse is an operation which returns the response from the GetFirmwareVersion operation.

ReplaceKeys is an operation to change the keys on an E-meter.

GetReplaceKeysResponse is an operation which returns the response from the ReplaceKeys operation.

UpdateFirmware is an operation to update the firmware module(s) on a device.

GetUpdateFirmwareResponse is an operation which returns the response from the UpdateFirmware operation.

GenerateAndReplaceKeys is an operation to generate and set the encryption and authentication key on a device.

GenerateAndReplaceKeysResponse is an operation which returns the response from the GenerateAndReplaceKeys operation.

SetClockConfiguration is an operation to set the clock configuration on a device.

GetSetClockConfigurationResponse is an operation which returns the response from the SetClockConfiguration operation.

ConfigureDefinableLoadProfile is an operation to configure the load profile on a device.

GetConfigureDefinableLoadProfileResponse is an operation which returns the response from the ConfigureDefinableLoadProfile operation.

SmartMeteringInstallation

AddDevice is an operation to add a device to the OSGP database.

GetAddDeviceResponse is an operation which returns the response from the AddDevice operation.

CoupleMbusDevice is an operation to couple an M-Bus device to a gateway.

GetCoupleMbusDeviceResponse is an operation which returns the response from the CoupleMbusDevice operation.

DeCoupleMbusDevice is an operation to decouple an M-Bus device from a gateway.

GetDeCoupleMbusDeviceResponse is an operation which returns the response from the DeCoupleMbusDevice operation.

SmartMeteringManagement

FindEvents is an operation to retrieve events logging from a device.

GetFindEventsResponse is an operation which returns the response from the FindEvents operation.

GetDevices is an operation to retrieve the last known relay statuses for a group of devices.

EnableDebugging is an operation to enable debug logging for a device.

GetEnableDebuggingResponse is an operation which returns the response from the EnableDebugging operation.

DisableDebugging is an operation to disable debug logging for a device.

GetDisableDebuggingResponse is an operation which returns the response from the DisableDebugging operation.

GetMessageLogs is an operation to read the debug logging from a device.

GetMessageLogsResponse is an operation which returns the response from the GetMessageLogs operation.

SetDeviceCommunicationSettings is an operation to set the OSGP device communication settings for a specific
SetDeviceCommunicationSettingsResponse is an operation which returns the response from the SetDeviceCommunicationSettings operation.

SetDeviceLifecycleStatus is an operation to set the lifecycle status from a device.

SetDeviceLifecycleStatusResponse is an operation which returns the response from the SetDeviceLifecycleStatus operation.

SetDeviceLifecycleStatusByChannel is an operation to set the lifecycle status from a device.

SetDeviceLifecycleStatusByChannelResponse is an operation which returns the response from the SetDeviceLifecycleStatusByChannel operation.

SmartMetering Monitoring

- GetActualMeterReads is an operation to retrieve the actual meter reads from an E-meter.
- GetActualMeterReadsResponse is an operation which returns the response from the ActualMeterReads operation.
- GetActualMeterReadsGas is an operation to retrieve the actual meter reads from a G-meter.
- GetActualMeterReadsGasResponse is an operation which returns the response from the ActualMeterReadsGas operation.
- GetPeriodicMeterReads is an operation to retrieve the periodic meter reads from an E-meter.
- GetPeriodicMeterReadsResponse is an operation which returns the response from the PeriodicMeterReads operation.
- GetPeriodicMeterReadsGas is an operation to retrieve the periodic meter reads from a G-meter.
- GetPeriodicMeterReadsGasResponse is an operation which returns the response from the PeriodicMeterReadsGas operation.
- GetProfileGenericData is an operation to retrieve any Profile Generic data from an E-meter.
- GetProfileGenericDataResponse is an operation which returns the response from the ProfileGenericData operation.
- ReadAlarmRegister is an operation to read the alarm register from a device.
- GetReadAlarmRegisterResponse is an operation which returns the response from the ReadAlarmRegister operation.
- RetrievePushNotificationAlarm is an operation to push retrieved alarm notifications to OSGP.
- ClearAlarmRegister is an operation to clear the Alarm register flags for pushed event notifications.
- ClearAlarmRegisterResponse is an operation which returns the response from the ClearAlarmRegister operation.

Device Management

- SetDeviceLifecycleStatus is an operation to set the device lifecycle status of a device.
- SetDeviceLifecycleStatusResponse is an operation which returns the response from the SetDeviceLifecycleStatus operation.

SmartMetering Notification

- SendNotification is an operation to let Webapps know there is a result ready to retrieve from the platform.

SmartMetering Bundle

- Bundle is a special operation in which one or more single operation(s) to a specific device can be bundled.
- GetBundleResponse is an operation which gets the response from the Bundle operation.

All operations sent to this device make use of one communication channel, which may improve performance considerably.

WSDL’s

- SmartMetering WSDL’s
- SmartMetering XSD schema's
Web Services

Smart Metering Web Services

This chapter describes all the web services in the smart metering domain.
bypass retry

By adding an element BypassRetry in namespace http://www.opensmartgridplatform.org/schemas/common/2014/10 with a value true or false, you can bypass retry when request to the device fail.
priority

By adding an element MessagePriority in namespace http://www.opensmartgridplatform.org/schemas/common/2014/10 with a value from 0 - 9 you can give a message a lower or higher priority.
scheduling

By adding an element ScheduleTime in namespace http://www.opensmartgridplatform.org/schemas/common/2014/10 with an xsd:dateTime value can schedule a message.
AdHocManagement

AdHocManagement

Describes the actions as defined in SmartMeteringAdhoc.wsdl
GetAssociationLnObjects

GetAssociationLnObjects request

Description
GetAssociationLnObjects is a request to get the Association LN object tree from an E-meter. The request is sent with the DeviceIdentification number from the desired device.

All requests have similar response behaviour which is described in ResponseMessages.

GetGetAssociationLnObjectsResponse returns the result values from getting the Association LN object. The response contains the DeviceIdentification and CorrelationUid which is received from the GetAssociationLnObjects request.

References
XSD: sm-adhoc.xsd
WSDL: SmartMeteringAdhoc.wsdl
GetGetAssociationLnObjectsResponse

GetGetAssociationLnObjectsResponse request

Description

GetGetAssociationLnObjectsResponse returns the result values from getting the Association LN object. The response contains the DeviceIdentification and CorrelationUid which is received from the GetAssociationLnObjects request.

All requests have similar response behaviour which is described in ResponseMessages.

References

XSD: sm-adhoc.xsd
WSDL: SmartMeteringAdhoc.wsdl
SpecificConfigurationObject

SpecificConfigurationObject request

Description
SpecificConfigurationObject is a request to retrieve the data for a specific configuration object indicated with:

- ClassId
- Attribute
- ObisCode

References
XSD: sm-configuration.xsd
WSDL: SmartMeteringBundle.wsdl
SynchronizeTime

SynchronizeTime request

Description

SynchronizeTime request synchronizes the date and time on a device. The date and time are retrieved from the server and sent to the device with CLASS_ID 8, OBIS_CODE 0.0.1.0.0.255 and ATTRIBUTE_ID 2. The request is sent with the DeviceIdentification number from the desired device. The request should contain a Deviation of local time to UTC in minutes (from the range of -720 to 720 inclusive) and a value Dst indicating whether daylight savings is active. For example in Central European Summer Time, DST is active and times are UTC/GMT +2 hours. For devices in a region where CEST applies, during the summer time the value for deviation should be “-120” (120 minutes deducted from local time gives GMT/UTC time) and dst should be "true".

All requests have similar response behaviour which is described in ResponseMessages.

GetSynchronizeTimeResponse returns the result from synchronizing date and time. The response contains the DeviceIdentification and CorrelationUid which is received from the SynchronizeTime request.

References

XSD: sm-adhoc.xsd

WSDL: SmartMeteringAdhoc.wsdl
**GetSynchronizeTimeResponse**

**GetSynchronizeTimeResponse request**

**Description**

GetSynchronizeTimeResponse returns the result from synchronizing date and time. The response contains the DeviceIdentifier and CorrelationUid which is received from the SynchronizeTime request.

All requests have similar response behaviour which is described in ResponseMessages.

**References**

XSD: [sm-adhoc.xsd](#)

WSDL: [SmartMeteringAdhoc.wsdl](#)
Bundle

Bundling

You can combine multiple requests to a meter in a bundle by creating a BundleRequest with one or more Actions in the namespace http://www.opensmartgridplatform.org/schemas/smartmetering/sm-bundle/2014/10. Each Action contains one of the existing requests to a meter.

A bundle is executed using one connection to the meter. A bundle response contains all individual responses of executed commands both successful and unsuccessful. When an individual request fails it is retried when this is useful, more precisely the bundle is retried, executing only requests that are fit for re-submission.
Bundle

Bundle request

Description

Bundle is a special request in which one or more single request(s) to a specific device can be bundled. All requests sent to this device make use of one communication channel, which may improve performance considerably.

GetBundleResponse returns the result of the actions of the bundle. The response contains the DeviceIdentification and CorrelationUid which is received from the Bundle request.

The Bundle request has an Actions tag. This contains a list of one or more single request(s). The response behavior is described in ResponseMessages.

Actions

Currently, the following actions are supported:

- FindEventsRequest see FindEvents
- SetSpecialDaysRequest see SetSpecialDays
- ReadAlarmRegisterRequest see ReadAlarmRegister
- GetActualMeterReadsRequest see GetActualMeterReadsGas
- GetAdministrativeStatusRequest see GetAdministrativeStatus
- GetPeriodicMeterReadsRequest see GetPeriodicMeterReads
- GetPeriodicMeterReadsGasRequest see GetPeriodicMeterReadsGas
- SetAdministrativeStatusRequest see SetAdministrativeStatus
- SetActivityCalendarRequest see SetActivityCalendar
- SetEncryptionKeyExchangeOnGMeterRequest see SetEncryptionKeyExchangeOnGMeter
- SetAlarmNotificationsRequest see SetAlarmNotifications
- SetConfigurationObjectRequest see SetConfigurationObject
- SetPushSetupAlarmRequest see SetPushSetupAlarm
- SetPushSetupSmsRequest see SetPushSetupSms
- SynchronizeTimeRequest see SynchronizeTime
- GetConfigurationRequest
- GetFirmwareVersionRequest see UpdateFirmware
- GetSpecificConfigurationObjectRequest see SpecificConfigurationObject
- SetKeysRequest
- GetAssociationLnObjectsRequest
- SetClockConfigurationRequest see SetClockConfiguration
- GetProfileGenericDataRequest see GetProfileGenericData
- ConfigureDefinableLoadProfileRequest see ConfigureDefinableLoadProfile
- SetMbusUserKeyByChannelRequest see SetMbusUserKeyByChannel
- GetMbusEncryptionKeyStatusRequest see GetMbusEncryptionKeyStatus

References

XSD: sm-bundle.xsd
WSDL: SmartMeteringBundle.wsdl
GetBundleResponse

GetBundleResponse request

Description
GetBundleResponse returns the result of the bundle requested with the Bundle method. The response contains the DeviceIdentification and CorrelationUid which is received from the Bundle request.

All requests have similar response behaviour which is described in ResponseMessages.

References
XSD:
- sm-adhoc.xsd
- sm-bundle.xsd
- sm-configuration.xsd
- sm-management.xsd
- sm-monitoring.xsd

WSDL: SmartMeteringBundle.wsdl
Configuration

Configuration

Describes the actions as defined in SmartMeteringConfiguration.wsdl
GetAdministrativeStatus

GetAdministrativeStatus request

Description
GetAdministrativeStatus is a request to retrieve the current AdministrativeStatus setting from a device. The request needs the DeviceIdentification.

All requests have similar response behaviour which is described in ResponseMessages. 

GetGetAdministrativeStatusResponse returns if the setting GetAdministrativeStatus is enabled. The response contains the DeviceIdentification and CorrelationUid which is received from the GetAdministrativeStatus request.

References
XSD: sm-configuration.xsd
WSDL: SmartMeteringConfiguration.wsd1
GetGetAdministrativeStatusResponse

GetGetAdministrativeStatusResponse request

Description

GetGetAdministrativeStatusResponse returns if the setting GetAdministrativeStatus is enabled. The response contains the DeviceIdentification and CorrelationUid which is received from the GetAdministrativeStatus request.

All requests have similar response behaviour which is described in ResponseMessages.

References

XSD: sm-configuration.xsd

WSDL: SmartMeteringConfiguration.wsdl
GetFirmwareVersion

GetFirmwareVersion request

Description

GetFirmwareVersion is a request to retrieve the firmware version(s) of a device. The request needs the DeviceIdentification.

All requests have similar response behaviour which is described in ResponseMessages.

GetGetFirmwareVersionResponse returns the version(s). The response contains the DeviceIdentification and CorrelationUid which is received from the GetFirmwareVersion request.

References

XSD: sm-configuration.xsd
WSDL: SmartMeteringConfiguration.wsdl
GetFirmwareResponse

GetFirmwareResponse request

Description

GetFirmwareResponse returns the device firmware versions requested with the GetFirmwareVersion method. The response contains the DeviceIdentification and CorrelationUid which is received from the GetFirmwareVersion request.

All requests have similar response behaviour which is described in ResponseMessages.

References

XSD: sm-configuration.xsd

WSDL: SmartMeteringConfiguration.wsdl
UpdateFirmware

UpdateFirmware request

Description

UpdateFirmware is a request to install another firmware version(s) on a device. The request needs the DeviceIdentification and the firmware versions, that together with the device model (as stored with the identified device) uniquely determine the firmware file to be used.

All requests have similar response behaviour which is described in ResponseMessages.

GetUpdateFirmwareResponse returns the version(s). The response contains the DeviceIdentification and CorrelationUid which is received from the UpdateFirmware request.

References

XSD: sm-configuration.xsd
WSDL: SmartMeteringConfiguration.wsdl
GetUpdateFirmwareResponse

GetUpdateFirmwareResponse request

Description

GetUpdateFirmwareResponse returns the device firmware versions that are on the device after calling the UpdateFirmware method. The response contains the DeviceIdentification and CorrelationUid which is received from the UpdateFirmware request.

All requests have similar response behaviour which is described in ResponseMessages.

References

XSD: sm-configuration.xsd

WSDL: SmartMeteringConfiguration.wsdl
ReplaceKeys

ReplaceKeys request

Description

ReplaceKeys is a request to change the keys on an E-meter. The request needs the DeviceIdentification, an AuthenticationKey and an EncryptionKey.

All requests have similar response behaviour which is described in ResponseMessages.

GetReplaceKeysResponse returns if the result is successful from the ReplaceKeys request. The response contains the DeviceIdentification and CorrelationUid which is received from the ReplaceKeys request.

References

XSD: sm-configuration.xsd

WSDL: SmartMeteringConfiguration.wsdl
GetReplaceKeysResponse

GetReplaceKeysResponse request

Description

GetReplaceKeysResponse returns if the result is successful from the ReplaceKeys request. The response contains the DeviceIdentification and CorrelationUid which is received from the ReplaceKeys request.

All requests have similar response behaviour which is described in ResponseMessages.

References

XSD: sm-configuration.xsd

WSDL: SmartMeteringConfiguration.wsdl
SetActivityCalendar

SetActivityCalendar request

Description

SetActivityCalendar is a request to set tariffs on an E-meter according a SeasonProfile and WeekProfile. In a WeekProfile, seven dayprofiles can be filled in with a start time and dayId which contains the tariff.

The request needs the DeviceIDentification, CalendarName, ActivatePassiveCalendarTime, SeasonProfileName, SeasonStart, WeekProfileName, DayId and StartTime.

All requests have similar response behaviour which is described in ResponseMessages.

GetSetActivityCalendarResponse returns the result from setting a SetActivityCalendar. The response contains the DeviceIDentification and CorrelationUid which is received from the SetActivityCalendar request.

References

XSD: sm-configuration.xsd

WSDL: SmartMeteringConfiguration.wsdl
GetSetActivityCalendarResponse

GetSetActivityCalendarResponse request

Description

GetSetActivityCalendarResponse returns the result from setting a SetActivityCalendar. The response contains the DeviceIdentification and CorrelationUid which is received from the SetActivityCalendar request.

All requests have similar response behaviour which is described in ResponseMessages.

References

XSD: sm-configuration.xsd

WSDL: SmartMeteringConfiguration.wSDL
SetAdministrativeStatus

SetAdministrativeStatus request

Description

SetAdministrativeStatus is a request to set the AdministrativeStatus on a device. The request needs the DeviceIdentification and Enabled parameter.

All requests have similar response behaviour which is described in ResponseMessages.

GetSetAdministrativeStatusResponse returns if the setting SetAdministrativeStatus is enabled. The response contains the DeviceIdentification and CorrelationUid which is received from the SetAdministrativeStatus request.

References

XSD: sm-configuration.xsd

WSDL: SmartMeteringConfiguration.wsdl
GetSetAdministrativeStatusResponse

GetSetAdministrativeStatusResponse request

Description

GetSetAdministrativeStatusResponse returns if the setting SetAdministrativeStatus is enabled. The response contains the DeviceIdentification and CorrelationUid which is received from the SetAdministrativeStatus request.

All requests have similar response behaviour which is described in ResponseMessages.

References

XSD: sm-configuration.xsd

WSDL: SmartMeteringConfiguration.wsdl
SetAlarmNotifications

SetAlarmNotifications request

Description
SetAlarmNotifications is a request to set the types of alarm notifications that must be notified from the device when they occur. The following notifications can be enabled or disabled:

- CLOCK_INVALID
- REPLACE_BATTERY
- POWER_UP
- PROGRAM_MEMORY_ERROR
- RAM_ERROR
- NV_MEMORY_ERROR
- MEASUREMENT_SYSTEM_ERROR
- WATCHDOG_ERROR
- FRAUD_ATTEMPT
- COMMUNICATION_ERROR_M_BUS_CHANNEL_1
- COMMUNICATION_ERROR_M_BUS_CHANNEL_2
- COMMUNICATION_ERROR_M_BUS_CHANNEL_3
- COMMUNICATION_ERROR_M_BUS_CHANNEL_4
- FRAUD_ATTEMPT_M_BUS_CHANNEL_1
- FRAUD_ATTEMPT_M_BUS_CHANNEL_2
- FRAUD_ATTEMPT_M_BUS_CHANNEL_3
- FRAUD_ATTEMPT_M_BUS_CHANNEL_4
- NEW_M_BUS_DEVICE_DISCOVERED_CHANNEL_1
- NEW_M_BUS_DEVICE_DISCOVERED_CHANNEL_2
- NEW_M_BUS_DEVICE_DISCOVERED_CHANNEL_3
- NEW_M_BUS_DEVICE_DISCOVERED_CHANNEL_4

The request needs the DeviceIdentification, AlarmType and Enabled parameters.

All requests have similar response behaviour which is described in ResponseMessages.

GetSetAlarmNotificationsResponse returns the result from setting a SetAlarmNotifications. The response contains the DeviceIdentification and CorrelationUid which is received from the SetAlarmNotifications request.

References

XSD: sm-configuration.xsd

WSDL: SmartMeteringConfiguration.wsdl
GetSetAlarmNotificationsResponse

**GetSetAlarmNotificationsResponse request**

**Description**

GetSetAlarmNotificationsResponse returns the result from setting a SetAlarmNotifications. The response contains the DeviceIdentification and CorrelationUid which is received from the SetAlarmNotifications request.

All requests have similar response behaviour which is described in ResponseMessages.

**References**

XSD: sm-configuration.xsd

WSDL: SmartMeteringConfiguration.wsdl
SetConfigurationObject

SetConfigurationObject request

Description

SetConfigurationObject is a request to set ConfigurationObject settings on a device. The attributes with OBIS code 0-1:94.31.3.255 give access to set GPRS_operation_mode setting and following flags:

- discover_on_open_cover
- discover_on_power_on
- dynamic_mbus_address
- P0_enable
- HLS_3_on_P3_enable
- HLS_4_on_P3_enable
- HLS_5_on_P3_enable
- HLS_3_on_P0_enable
- HLS_4_on_P0_enable
- HLS_5_on_P0_enable

See DSMR document chapter 8.3 for detailed description. The request needs the DeviceIdentification, GprsOperationMode, ConfigurationFlagType and Enabled parameters.

All requests have similar response behaviour which is described in ResponseMessages.

GetSetConfigurationObjectResponse returns the result from setting a SetConfigurationObject. The response contains the DeviceIdentification and CorrelationUid which is received from the SetConfigurationObject request.

References

XSD: sm-configuration.xsd
WSDL: SmartMeteringConfiguration.wsdl
GetSetConfigurationObjectResponse

GetSetConfigurationObjectResponse request

Description

GetSetConfigurationObjectResponse returns the result from setting a ConfigurationObject. The response contains the DeviceIdentification and CorrelationUid which is received from the SetConfigurationObject request.

All requests have similar response behaviour which is described in ResponseMessages.

References

XSD: sm-configuration.xsd

WSDL: SmartMeteringConfiguration.wsdl
SetEncryptionKeyExchangeOnGMeter

SetEncryptionKeyExchangeOnGMeter request

Description

SetEncryptionKeyExchangeOnGMeter is a request to transfer and set a G-meter key on a G-meter via the E-meter. The request needs the DeviceIdentification from the G-meter. If the device identification of the G-meter is not known, but the gateway device identification and M-Bus channel are known, use the SetMbusUserKeyByChannel request instead.

All requests have similar response behaviour which is described in ResponseMessages.

GetSetEncryptionKeyExchangeOnGMeterResponse returns the result from setting a SetEncryptionKeyExchangeOnGMeter. The response contains the DeviceIdentification and CorrelationUid which is received from the SetEncryptionKeyExchangeOnGMeter request.

References

XSD: sm-configuration.xsd
WSDL: SmartMeteringConfiguration.wsdl
GetSetEncryptionKeyExchangeOnGMeterResponse

GetSetEncryptionKeyExchangeOnGMeterResponse request

Description
GetSetEncryptionKeyExchangeOnGMeterResponse returns the result from setting a SetEncryptionKeyExchangeOnGMeter. The response contains the DeviceIdentification and CorrelationUid which is received from the SetEncryptionKeyExchangeOnGMeter request.

All requests have similar response behaviour which is described in ResponseMessages.

References
XSD: sm-configuration.xsd
WSDL: SmartMeteringConfiguration.wsdl
SetPushSetupAlarm

SetPushSetupAlarm request

Description

SetPushSetupAlarm is a request to define the destination of the TCP message that is optionally sent by the device. The request needs the DeviceIdentification, Host URL and port.

All requests have similar response behaviour which is described in ResponseMessages.

GetSetPushSetupAlarmResponse returns the result from setting a SetPushSetupAlarm. The response contains the DeviceIdentification and CorrelationUid which is received from the SetPushSetupAlarm request.

References

XSD: sm-configuration.xsd
WSDL: SmartMeteringConfiguration.wsdl
GetSetPushSetupAlarmResponse

Description

GetSetPushSetupAlarmResponse returns the result from setting a SetPushSetupAlarm. The response contains the DeviceIdentification and CorrelationUid which is received from the SetPushSetupAlarm request.

All requests have similar response behaviour which is described in ResponseMessages.

References

XSD: sm-configuration.xsd

WSDL: SmartMeteringConfiguration.wsdl
SetPushSetupSms

SetPushSetupSms request

Description

SetPushSetupSms is a request to set an endpoint in a device which tells the device where to connect to when it is woken. The request needs the DeviceIdentification, host URL and port.

All requests have similar response behaviour which is described in ResponseMessages.

GetSetPushSetupSmsResponse returns the result from setting a SetPushSetupSms. The response contains the DeviceIdentification and CorrelationUid which is received from the SetPushSetupSms request.

References

XSD: sm-configuration.xsd

WSDL: SmartMeteringConfiguration.wsdl
GetSetPushSetupSmsResponse

GetSetPushSetupSmsResponse request

Description

GetSetPushSetupSmsResponse returns the result from setting a SetPushSetupSms. The response contains the DeviceIdentification and CorrelationUid which is received from the SetPushSetupSms request.

All requests have similar response behaviour which is described in ResponseMessages.

References

XSD: sm-configuration.xsd

WSDL: SmartMeteringConfiguration.wsdl
SetSpecialDays

SetSpecialDays request

Description

SetSpecialDays is a request to set a dayId profile for a specific date on a device, other than the standard applicable dayId's. This can be useful to change tariffs and tariff scheduling for specific days such as public holidays. The request is send with the DeviceIdentification number from the desired device, date and dayId.

All requests have similar response behaviour which is described in ResponseMessages. GetSetSpecialDaysResponse returns the result from setting a Special Day. The response contains the DeviceIdentification and CorrelationUid which is received from the SetSpecialDays request.

References

XSD: sm-configuration.xsd
WSDL: SmartMeteringConfiguration.wsdl
GetSetSpecialDaysResponse

GetSetSpecialDaysResponse request

Description

GetSetSpecialDaysResponse returns the result from setting a Special Day. The response contains the DeviceIdentification and CorrelationUid which is received from the SetSpecialDays request.

All requests have similar response behaviour which is described in ResponseMessages.

References

XSD: sm-configuration.xsd

WSDL: SmartMeteringConfiguration.wsdl
GetConfigurationObject

GetConfigurationObject request

Description

GetConfigurationObject is a request to retrieve a ConfigurationObject from a device. The configuration object in the electricity meter with the OBIS code 0-1:94.31.3.255 is used to access the GPRS_operation_mode setting and following flags:

- discover_on_open_cover
- discover_on_power_on
- dynamic_mbus_address
- P0_enable
- HLS_3_on_P3_enable
- HLS_4_on_P3_enable
- HLS_5_on_P3_enable
- HLS_3_on_P0_enable
- HLS_4_on_P0_enable
- HLS_5_on_P0_enable

See DSMR document chapter 8.3 for detailed description. The request needs the DeviceIdentification.

All requests have similar response behavior which is described in ResponseMessages.

References

XSD: sm-configuration.xsd
WSDL: SmartMeteringConfiguration.wsdl
GetConfigurationObjectResponse

GetConfigurationObjectResponse request

Description
GetConfigurationObjectResponse returns the result, a ConfigurationObject, which is received from the GetConfigurationObject request.

All requests have similar response behavior which is described in ResponseMessages.

References
XSD: sm-configuration.xsd
WSDL: SmartMeteringConfiguration.wsdl
ConfigureDefinableLoadProfile

ConfigureDefinableLoadProfile request

Description

ConfigureDefinableLoadProfile is a request to change the configuration of the definable load profile (COSEM object of interface class 'Profile generic' with logical name '0-1:94.31.6.255') of the device. The request needs the DeviceIdentification, and at least one of CaptureObjects and CapturePeriod.

The CaptureObjects element may be included in the request to specify one or more objects to be captured in the definable load profile, containing definitions as CaptureObject according to the CaptureObjectDefinition in common.xsd. The CaptureObjects should not include the clock definition ([8,0-0:1.0.0.255,2,0]) as this will always be included as first capture object. This matches the way GetProfileGenericData works when retrieving the buffer of the definable load profile (where you must not specify the clock definition as selected value).

The CapturePeriod may be included to specify the automatic capturing period in seconds (a value of zero meaning no automatic capturing should be done).

All requests have similar response behaviour which is described in ResponseMessages.

The response contains the DeviceIdentification and CorrelationUid which is received from the ConfigureDefinableLoadProfile request. GetConfigureDefinableLoadProfileResponse returns if the result is successful from the ConfigureDefinableLoadProfile request.

References

XSD: sm-configuration.xsd
WSDL: SmartMeteringConfiguration.wsdl
GetConfigureDefinableLoadProfileResponse

GetConfigureDefinableLoadProfileResponse request

Description

GetConfigureDefinableLoadProfileResponse returns if the result is successful from the ConfigureDefinableLoadProfile request. The request contains the DeviceIdentification and CorrelationUid which is received from the ConfigureDefinableLoadProfile request.

All requests have similar response behaviour which is described in ResponseMessages.

References

XSD: sm-configuration.xsd
WSDL: SmartMeteringConfiguration.wsdl
SetMbusUserKeyByChannel

SetMbusUserKeyByChannel request

Description
SetMbusUserKeyByChannel is a request to generate, transfer and set an M-Bus user key on an M-Bus device (for instance a G-meter behind an E-meter) via the DLMS gateway device. The request needs the DeviceIdentification from the gateway device and the channel for the M-Bus device. A use case for a request with the channel (as only identification of the M-Bus device besides the identification of the gateway) as input is to be able to respond to new M-Bus device discovered on channel x alarms (x in 1..4) from a gateway. If a new M-Bus User key is to be set on an M-Bus device with a known identification, this can be done with the SetEncryptionKeyExchangeOnGMeter request.

All requests have similar response behaviour which is described in ResponseMessages.

The response contains the DeviceIdentification and CorrelationUid which is received from the SetMbusUserKeyByChannel request. GetSetMbusUserKeyByChannelResponse returns the result from issuing a SetMbusUserKeyByChannel request.

References

XSD: sm-configuration.xsd
WSDL: SmartMeteringConfiguration.wsdl
GetSetMbusUserKeyByChannelResponse

GetSetMbusUserKeyByChannelResponse request

Description

GetSetMbusUserKeyByChannelResponse returns the result from issuing a SetMbusUserKeyByChannel request. The request contains the DeviceIdentification and CorrelationUid which is received from the SetMbusUserKeyByChannel request.

All requests have similar response behaviour which is described in ResponseMessages.

References

XSD: sm-configuration.xsd

WSDL: SmartMeteringConfiguration.wsdl
GetMBusEncryptionKeyStatus

GetMBusEncryptionKeyStatus request

Description

GetMBusEncryptionKeyStatus is a request to retrieve the encryption key status of a M-Bus device from an E-meter. The request needs the DeviceIdentification of the M-Bus Device.

All requests have similar response behaviour which is described in ResponseMessages.

The returned response for the GetMBusEncryptionKeyStatus request is as specified in GetGetMBusEncryptionKeyStatusResponse.

References

XSD: sm-configuration.xsd

WSDL: SmartMeteringConfiguration.wsdl
GetGetMbusEncryptionKeyStatusResponse

GetGetMBusEncryptionKeyStatusResponse request

Description

GetGetMbusEncryptionKeyStatusResponse is a request to return the M-Bus encryption key status as requested by a GetMbusEncryptionKeyStatus request. The possible return values for the M-Bus encryption key status can be found in the EncryptionKeyStatus enum in the sm-configuration.xsd

References

XSD: sm-configuration.xsd
WSDL: SmartMeteringConfiguration.wsdl
GetMbusEncryptionKeyStatusByChannel

GetMBusEncryptionKeyStatusByChannel request

Description

GetMbusEncryptionKeyStatusByChannel is a request to retrieve the encryption key status of an M-Bus device from an E-meter. The request needs the DeviceIdentification of the gateway device and a channel.

All requests have similar response behaviour which is described in ResponseMessages.

The returned response for the GetMbusEncryptionKeyStatusByChannel request is as specified in GetMbusEncryptionKeyStatusByChannelResponse.

References

XSD: sm-configuration.xsd
WSDL: SmartMeteringConfiguration.wsdl
ScanMbusChannels

ScanMbusChannels request

Description

ScanMbusChannels is a request to read the M-Bus Short Equipment Identifier (Short ID) attributes (Identification number, Manufacturer identification, Version identification, and Device type identification) from all four channels on a Gateway device to determine if an M-Bus device is bound on a channel of the Gateway device.

All requests have similar response behaviour which is described in ResponseMessages.

The returned response for the ScanMbusChannels request is as specified in ScanMbusChannelsResponse.

References

XSD: sm-management.xsd

WSDL: SmartMeteringAdhoc.wsdl
ScanMbusChannelsResponse

ScanMbusChannelsResponse request

Description

ScanMbusChannelsResponse returns the result of a ScanMbusChannels request. The response contains the M-Bus Short Equipment Identifier (Short ID) attributes (Identification number, Manufacturer identification, Version identification, and Device type identification) from all four channels of a Gateway device.

References

XSD: sm-adhoc.xsd
WSDL: SmartMeteringAdhoc.wsdl
Installation

Installation

Describes the actions as defined in SmartMeteringInstallation.wsdl
AddDevice

AddDevice request

Description
AddDevice is a request to add a device to the OSGP database. For the list of parameters, see the .xsd file (link below).

All requests have similar response behaviour which is described in ResponseMessages. GetAddDeviceResponse returns if the result is successful from the request. The response contains the DeviceIdentification and CorrelationUid which is received from the AddDevice request.

References
XSD: sm-installation.xsd
WSDL: SmartMeteringInstallation.wsdl

Example scenario
Scenario: Add a new device
When receiving a smartmetering add device request
| DeviceIdentification | TEST1024000000001 |
| DeviceType           | SMART_METER_E   |
| CommunicationMethod  | GPRS           |
| CommunicationProvider| KPN            |
| ICC_id               | 1234           |
| DSMR_version         | 4.2.2          |
| Supplier             | Kaifa          |
| HLS3_active          | false          |
| HLS4_active          | false          |
| HLS5_active          | true           |
| Master_key           | m_key          |
| Authentication_key   | a_key          |
| Encryption_key       | e_key          |

Then the get add device response should be returned
| DeviceIdentification | TEST1024000000001 |
| Result               | OK              |

And the dlns device with identification "TEST1024000000001" exists
And a request to the device can be performed after activation
And the stored keys are not equal to the received keys

Example XML Message
  <soapenv:Header>
    <ns:ApplicationName>AutomaticTest</ns:ApplicationName>
    <ns:UserName>SoapUI</ns:UserName>
    <ns:OrganisationIdentification>test.org</ns:OrganisationIdentification>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:AddDeviceRequest>
      <ns1:Device>
        <!--Optional:-->
        <ns1:Device_Identification>TEST1024000000001</ns1:Device_Identification>
        <!--Optional:-->
        <ns1:Device_type>SMART_METER_E</ns1:Device_type>
        <!--Optional:-->
        <ns1:Communication_method>GPRS</ns1:Communication_method>
        <!--Optional:-->
        <ns1:Communication_provider>KPN</ns1:Communication_provider>
        <!--Optional:-->
        <ns1:ICC_id>icc_id</ns1:ICC_id>
        <!--Optional:-->
        <ns1:DSMR_version>4.2.2</ns1:DSMR_version>
        <!--Optional:-->
        <ns1:Supplier>Kaifa</ns1:Supplier>
        <!--Optional:-->
        <ns1:HLS3_active>false</ns1:HLS3_active>
      </ns1:Device>
    </ns1:AddDeviceRequest>
  </soapenv:Body>
</soapenv:Envelope>
<ns1:HLS4_active>false</ns1:HLS4_active>
<!--Optional:-->
<ns1:HLS5_active>true</ns1:HLS5_active>
<!--Optional:-->
<ns1:Master_key>m_key</ns1:Master_key>
<!--Optional:-->
<ns1:Global_encryption_unicast_key>e_key</ns1:Global_encryption_unicast_key>
<!--Optional:-->
<ns1:Authentication_key>a_key</ns1:Authentication_key>
<ns1:Delivery_date>2017-05-17 06:16:57.07</ns1:Delivery_date>
</ns1:AddDeviceRequest>
</soapenv:Body>
</soapenv:Envelope>
GetAddDeviceResponse

GetAddDeviceResponse request

Description
GetAddDeviceResponse returns if the result is successful from the AddDevice request. The response contains the DeviceIdentification and CorrelationUid which is received from the AddDevice request.

All requests have similar response behaviour which is described in ResponseMessages.

References
XSD: sm-installation.xsd
WSDL: SmartMeteringInstallation.wsdl
CoupleMbusDevice

CoupleMbusDevice request

Description

CoupleMbusDevice is a request to couple a gateway and a m-bus device. The request needs the following parameters:

- DeviceIdentification
- MbusDeviceIdentification

All requests have similar response behaviour which is described in ResponseMessages.

GetCoupleMbusDeviceResponse returns if the result is successful from the request. The response request contains the DeviceIdentification and CorrelationUid which is received from the CoupleMbusDevice request.

References

XSD: sm-installation.xsd

WSDL: SmartMeteringInstallation.wsdl
GetCoupleMbusDeviceResponse

GetCoupleMbusDeviceResponse request

Description

GetCoupleMbusDeviceResponse returns if the result is successful from the CoupleMbusDevice request. The response contains the DeviceIdentification and CorrelationUid which is received from the CoupleMbusDevice request.

All requests have similar response behaviour which is described in ResponseMessages.

References

XSD: sm-installation.xsd

WSDL: SmartMeteringInstallation.wsdl
DeCoupleMbusDevice

DeCoupleMbusDevice request

Description

DeCoupleMbusDevice is a request to decouple an Mbus device (such as a gas meter) from a device to the OSGP database. The request needs the following parameters:

- DeviceIdentification
- MbusDeviceIdentification

All requests have similar response behaviour which is described in ResponseMessages. GetDeCoupleMbusDeviceResponse returns if the result is successful from the request. The response contains the DeviceIdentification and CorrelationUid which is received from the DeCoupleMbusDevice request.

References

XSD: sm-installation.xsd

WSDL: SmartMeteringInstallation.wsdl
GetDeCoupleMbusDeviceResponse

GetDeCoupleMbusDeviceResponse request

Description

GetDeCoupleMbusDeviceResponse returns if the result is successful from the DeCoupleMbusDevice request. The response contains the DeviceIdentification and CorrelationUid which is received from the DeCoupleMbusDevice request. All requests have similar response behaviour which is described in ResponseMessages.

References

XSD: sm-installation.xsd

WSDL: SmartMeteringInstallation.wsdl
Management

Management

Describes the actions as defined in SmartMeteringManagement.wsdl
FindEvents

FindEvents request

Description

FindEvents is a request to retrieve periodic events logging from a device. The request needs the DeviceIdentification, EventLogCategory, From and Until DateTime. The EventLogCategories consist of:

- STANDARD_EVENT_LOG
- FRAUD_DETECTION_LOG
- COMMUNICATION_SESSION
- M_BUS_EVENT_LOG

DSMR Chapter 4.2.1 describes the several events and their description.

All requests have similar response behaviour which is described in ResponseMessages.

GetFindEventsResponse returns if the result is successful from the request. The response contains the DeviceIdentification and CorrelationUid which is received from the FindEvents request.

References

XSD: sm-management.xsd

WSDL: SmartMeteringManagement.wsd1
GetFindEventsResponse

GetFindEventsResponse request

Description

GetFindEventsResponse returns if the result is successful from the FindEvents request. The response contains the DeviceIdentification and CorrelationUid which is received from the FindEvents request.

All requests have similar response behaviour which is described in ResponseMessages.

References

XSD: sm-management.xsd
WSDL: SmartMeteringManagement.wsdl
GetDevices

GetDevices request

Description

GetDevices is a request to get the last known relay statuses for a group of devices, so you can get an overview of statuses for a specific set of devices. The request needs the Page parameter.

All requests have similar response behaviour which is described in ResponseMessages.

References

XSD: sm-management.xsd

WSDL: SmartMeteringManagement.wsdl
SetDeviceLifecycleStatusByChannel

SetDeviceLifecycleStatusByChannel request

Description

SetDeviceLifecycleStatusByChannel is a request to set the device lifecycle status of an Mbus device, using the device identification of the Gateway device and a channel.

All requests have similar response behaviour which is described in ResponseMessages. The returned response for the SetDeviceLifecycleStatusByChannel request is as specified in SetDeviceLifecycleStatusByChannelResponse.

References

XSD: sm-management.xsd
WSDL: SmartMeteringManagement.wsdl
SetDeviceLifecycleStatusByChannelResponse

SetDeviceLifecycleStatusByChannelResponse request

Description

SetDeviceLifecycleStatusByChannelResponse returns the result of a SetDeviceLifecycleStatusByChannel request. The response contains the GatewayDeviceIdentification, MbusDeviceIdentification, DeviceLifecycleStatus and channel. SetDeviceLifecycleStatusByChannel request.

References

XSD: sm-management.xsd
WSDL: SmartMeteringManagement.wsdl
EnableDebugging

EnableDebugging request

Description
Enable debugging for a device. Communication with the device will be logged and made available through FindMessageLogs.

All requests have similar response behaviour which is described in ResponseMessages.

GetEnableDebuggingResponse returns the result status. The response contains the DeviceIdentification and CorrelationUid which is received from the GetEnableDebuggingRequest request.

References
XSD: sm-management.xsd
WSDL: SmartMeteringManagement.wsdl
DisableDebugging

DisableDebugging request

Description
Disable debugging for a device. Communication with the device will be logged and made available through FindMessageLogs.

All requests have similar response behaviour which is described in ResponseMessages.

GetDisableDebuggingResponse returns the result status. The response contains the DeviceIdentification and CorrelationUid which is received from the GetDisableDebuggingRequest request.

References
XSD: sm-management.xsd
WSDL: SmartMeteringManagement.wsdl
FindMessageLogs

FindMessageLogs request

Description
FindMessageLogs is a request to retrieve logged messages for a device. The request needs the DeviceIdentification and a Page number to return.

All requests have similar response behaviour which is described in ResponseMessages.

GetFindMessageLogsResponse returns if the result is successful from the request. The response contains the DeviceIdentification and CorrelationUid which is received from the FindMessageLogs request.

Note: This functionality also exists in the admin device management service. It was duplicated here to be implemented asynchronously, as there is no support for asynchronous requests triggering a notification service in the admin project. As soon as asynchronous requests and notifications are implemented throughout the OSGP platform, this method should be removed in favour of the admin implementation.

References

XSD: sm-management.xsd
WSDL: SmartMeteringManagement.wsdl
Monitoring

Monitoring

Describes the functions as defined in SmartMeteringMonitoring.wsdl
GetActualMeterReads

GetActualMeterReads request

Description
GetActualMeterReads is a request to retrieve the actual import and export meter reads from an E-meter. The request needs the DeviceIdentification.

All requests have similar response behaviour which is described in ResponseMessages.

GetActualMeterReadsResponse returns the retrieved meter reads values, unit and log time from the GetActualMeterReads request. The response contains the DeviceIdentification and CorrelationUid which is received from the GetActualMeterReads request.

References
XSD: sm-monitoring.xsd
WSDL: SmartMeteringMonitoring.wsd
GetActualMeterReadsResponse

GetActualMeterReadsResponse request

Description

GetActualMeterReadsResponse returns the retrieved import and export values, unit and logtime from the ActualMeterReads request. The response contains the DeviceIdentification and CorrelationUid which is received from the GetActualMeterReads request.

All requests have similar response behaviour which is described in ResponseMessages.

References

XSD: sm-monitoring.xsd
WSDL: SmartMeteringMonitoring.wsdl
GetActualMeterReadsGas

GetActualMeterReadsGas request

Description

GetActualMeterReadsGas is a request to retrieve the actual import and export meter reads from a G-meter. The request needs the DeviceIdentification.

All requests have similar response behaviour which is described in ResponseMessages.

GetActualMeterReadsGasResponse returns the retrieved meter reads values, unit and log time from the GetActualMeterReadsGas request. The response contains the DeviceIdentification and CorrelationUid which is received from the GetActualMeterReadsGas request.

References

XSD: sm-monitoring.xsd
WSDL: SmartMeteringMonitoring.wsd
GetActualMeterReadsGasResponse

GetActualMeterReadsGasResponse request

Description

GetActualMeterReadsGasResponse returns the retrieved import and export values, unit and log time from the ActualMeterReadsGas request. The response contains the DeviceIdentification and CorrelationUid which is received from the ActualMeterReadsGas request.

All requests have similar response behaviour which is described in ResponseMessages.

References

XSD: sm-monitoring.xsd

WSDL: SmartMeteringMonitoring.wsd
GetPeriodicMeterReads

GetPeriodicMeterReads request

Description

GetPeriodicMeterReads is a request to retrieve the periodic import and export meter reads from an E-meter. The period can be DAILY, MONTHLY or INTERVAL. The request needs the DeviceIdentification.

All requests have similar response behaviour which is described in ResponseMessages.

GetPeriodicMeterReadsResponse returns the retrieved meter reads values, unit and log time from the GetPeriodicMeterReads request. The response contains the DeviceIdentification and CorrelationUid which is received from the GetPeriodicMeterReads request.

References

XSD: sm-monitoring.xsd
WSDL: SmartMeteringMonitoring.wsd
GetPeriodicMeterReadsResponse

GetPeriodicMeterReadsResponse request

Description

GetPeriodicMeterReadsResponse returns the retrieved import and export values, unit and log time from the PeriodicMeterReads request. The response contains the DeviceIdentification and CorrelationUid which is received from the GetPeriodicMeterReads request.

All requests have similar response behaviour which is described in ResponseMessages.

References

XSD: sm-monitoring.xsd

WSDL: SmartMeteringMonitoring.wsd1
GetPeriodicMeterReadsGas

GetPeriodicMeterReadsGas request

Description

GetPeriodicMeterReadsGas is a request to retrieve the periodic import and export meter reads from a G-meter. The period can be DAILY, MONTHLY or INTERVAL. The request needs the DeviceIdentification.

All requests have similar response behaviour which is described in ResponseMessages.

GetPeriodicMeterReadsGasResponse returns the retrieved meter reads values, unit and log time from the GetPeriodicMeterReadsGas request. The response contains the DeviceIdentification and CorrelationUid which is received from the GetPeriodicMeterReadsGas request.

References

XSD: sm-monitoring.xsd
WSDL: SmartMeteringMonitoring.wsd1
GetPeriodicMeterReadsGasResponse

GetPeriodicMeterReadsGasResponse request

Description

GetPeriodicMeterReadsGasResponse returns the retrieved import and export values, unit and log time from the PeriodicMeterReadsGas request. The response contains the DeviceIdentification and CorrelationUid which is received from the GetPeriodicMeterReadsGas request.

All requests have similar response behaviour which is described in ResponseMessages.

References

XSD: sm-monitoring.xsd

WSDL: SmartMeteringMonitoring.wsdl
GetProfileGenericData

GetProfileGenericData request

Description

GetProfileGenericData is a request to retrieve any DLMS "Profile generic" data from an E-meter. The request needs the DeviceIdentification.

The specific Profile generic data to be retrieved is identified by its OBIS code included as ObisCode according to the ObisCodeValues as specified in common.xsd.

Selective access will be applied as described in the DLMS standard for access selector range_descriptor. The clock definition is used as restricting_object. The from_value and to_value for the captured clock values will be set based on the BeginDate and EndData in the request.

It is possible to further reduce the amount of data retrieved from the device to specify selected_values. This is done by including the optional SelectedValues element in the request specifying one or more capture object definitions as CaptureObject according to the CaptureObjectDefinition in common.xsd.

The clock definition must not be specified in the SelectedValues, since it will always be included in the results. The values that are specified must be capture object definitions that appear in the list of capture_objects for the Profile generic data that is retrieved.

All requests have similar response behaviour which is described in ResponseMessages.

The ultimately returned response for the GetProfileGenericData request is as specified in GetProfileGenericDataResponse.

References

XSD: sm-monitoring.xsd

WSDL: SmartMeteringMonitoring.wsd
GetProfileGenericDataResponse

GetProfileGenericDataResponse request

Description

GetProfileGenericDataResponse is a request to return the Generic profile buffer data as requested by a GetProfileGenericData request. It contains the DeviceIdentification and CorrelationUid which is received from the GetProfileGenericData request.

The response to the GetProfileGenericDataResponse request contains the logical name of the requested Generic profile as LogicalName according to the ObisCodeValues as specified in common.xsd.

The definitions of the capture objects from the buffer that are included in the response are listed as CaptureObject according to the CaptureObject in common.xsd.

The actual data from the buffer is included in the ProfileEntries, where each ProfileEntry has a list of values that match the capture objects from the response.

References

XSD: sm-monitoring.xsd

WSDL: SmartMeteringMonitoring.wsdl
ReadAlarmRegister

ReadAlarmRegister request

Description
ReadAlarmRegister is a request to retrieve the query alarm register. A notification will be sent and the query will be stored in the database. The request needs the DeviceIdentification.

All requests have similar response behaviour which is described in ResponseMessages.

GetReadAlarmRegisterResponse returns the alarm notifications from the ReadAlarmRegister request. The response contains the DeviceIdentification and CorrelationUid which is received from the ReadAlarmRegister request.

References
XSD: sm-monitoring.xsd
WSDL: SmartMeteringMonitoring.wsdl
GetReadAlarmRegisterResponse

GetReadAlarmRegisterResponse request

Description

GetReadAlarmRegisterResponse returns the alarm notifications from the ReadAlarmRegister request. The response contains the DeviceIdentification and CorrelationUid which is received from the ReadAlarmRegister request.

All requests have similar response behaviour which is described in ResponseMessages.

References

XSD: sm-monitoring.xsd

WSDL: SmartMeteringMonitoring.wsd1
RetrievePushNotificationAlarm

RetrievePushNotificationAlarm request

Description
RetrievePushNotificationAlarm is a request to push retrieved alarm notifications to OSGP. The request needs the DeviceIdentification.

All requests have similar response behaviour which is described in ResponseMessages.

References
XSD: sm-monitoring.xsd
WSDL: SmartMeteringMonitoring.wSDL
Notification

Notifications

Describes the actions as defined in the SmartMeteringNotification.wsdl
SendNotification

SendNotification request

Description

SendNotification is a request from the platform to let Webapps know there is a result ready to retrieve. In this way, there is no need for constant polling between Webapps and the platform. The request needs the DeviceIdentification.

All requests have similar response behaviour which is described in ResponseMessages.

References

XSD: sm-notification.xsd

WSDL: SmartMeteringNotification.wsdl
The response of a request should always contain a DeviceIdentification and CorrelationUid which is used in the response request. Assertions validate if there is a 'SOAP Response' received, if the response is 'Schema Compliant' with the WSDL and if there has been a 'not SOAP Fault'. The last one occurs when a fault code is returned. Possible faults are connection timeout, SmartMeter could not be found, TCP-IP connection error or CorrelationUid is unknown. The faults can be among others a FunctionalFault or TechnicalFault. Responses to a bundle request may include faults in the form of FaultResponseData, resembling the other faults. The format of these faults is described in the common.xsd.
Use cases

Example use-case for this domain

Up-to-date information on use-cases can be found on the Grid eXchange Fabric website.

Smart Meter Head-end System

Technical drivers

- Replacement or addition to the current head-end system
- During the coming years, many smart meters will be placed in houses, companies and other properties, therefore grid operators need a scalable solution
- E(lectricity) Meters can host up to 4 other smart devices, Gas Meters for example
- DLMS/COSEM is used by many (if not all) Smart Meters

Customer drivers

- People will have more insight in their power consumption
- Meter values can be gathered by the grid operator, instead of relying on people reporting the meter values
Guidelines to add a new domain to GXF

Guidelines to add a new domain to OSGP

In order to add a new domain to OSGP, you can benefit from the guidelines given in this document. The general idea for adding a new domain is to copy an existing domain, for instance the microgrids domain, and perform a global search and replace, to replace the old domain name with the new domain name. You can use refactor methods from IntelliJ or Eclipse to help renaming the old domain names.

To add a new domain, changes must be made to 2 GitHub repositories:

1. Config repository
2. Open smart grid platform repository

Changes to OSGP/Config

Search for “Microgrids” and “microgrids” in all files and you will find all files to change for a new domain. These files include:

- Apache configuration
- Create domain database script
- Backup, restore, symlinks scripts
- Tomcat context script

Changes to OSGP/open-smart-grid-platform

Directory OSGP/open-smart-grid-platform/osgp/shared/

A new Maven module must be added for the new domain (osgp-ws-newdomain). This module will contain the wsdl files for the new domain services. Copy for instance osgp-ws-microgrids, search and replace microgrids with your domain and replace the wsdl files with your wsdl files. JAXB will generate java classes for your webservices. Change NewDomainWebServiceConfig.java accordingly. Make sure your base Request and Response classes are generated with an @XmlRootElement annotation. Otherwise your endpoints which are based on these types will fail (See @PayloadRoot in AdHocManagementEndpoint in osgp-adapter-ws-microgrids). The structure of your wsdl file determines whether the @XmlRootElement annotation is generated or not.

Image showing the generated @XmlRootElement annotation

```java
package org.opensmartgridplatform.adapter.ws.schema.microgrids.adhocmanagement;

import java.util.ArrayList;

@XmlElementType(name = "Microgrids", propOrder = {
    "DeviceIdentification",
    "system"
})
@XmlElement(name = "GetDeviceIdentification")
public class GetDeviceIdentification {

    @XmlElement(name = "DeviceIdentification", required = true)
    @XmlAccessorType(XmlAccessType.FIELD)
    protected String deviceIdentification;

    @XmlElement(name = "System")
    protected List<SystemFilter> system;

    /**
     * Gets the value of the deviceIdentification property.
     *
     * @return possible object is [null String]
     */

    Do not forget to add two constants for your new domain in the enum ComponentType.java (https://github.com/OSGP/open-smart-grid-platform/blob/development/osgp/shared/shared/src/main/java/org/opensmartgridplatform/shared/exceptionhandling/ComponentType.java).
Add DTO’s to osgp-dto for your services. The DTO’s are used in the protocol-adapter. Mapping from/to DTO’s is performed in adapter-domain.

Directory OSGP/open-smart-grid-platform/osgp/platform/

Reference the new osgp-ws-newdomain in pom.xml. Also create three new Maven modules and add them to the pom:

- osgp-domain-newdomain
- osgp-adapter-ws-newdomain
- osgp-adapter-domain-newdomain

Constants for the new domain we services

OSGP uses a couple of Java enums to identify all available services the platform offers.

- The DeviceFunction enum contains all services for all domains.
- The NotificationType enum and the NewDomainRequestMessageType enum are identical and contain the services for 1 domain. The NotificationType enum is generated from the wsdl service definition for the notification service. The NewDomainType enum is defined in the Web Service Layer for the new domain and is used to pass the message type to the other layers of OSGP.
- DeviceRequestMessageType will contain the services for 1 protocol. Strictly speaking this enum is not necessary to add a new domain because the enum is located in the protocol layer of OSGP.

Each new service that is offered by the domain, for instance GET_DATA or SET_DATA, must be added to 3 java enums:


SQL for the new domain

A Flyway script should be added for system data. For a new domain a new record must be inserted in the table domain_info in the core database. Check for instance the Flyway script for Distribution Automation https://github.com/OSGP/open-smart-grid-platform/blob/development/osgp/platform/osgp-core/src/main/resources/db/migration/V20170508125704045__Added_Distribution_Automation_domain_info.sql. Test data for a new domain will include:

- Table device_function_mapping in the core database. Add a row for each new service to authorize ‘OWNER’ for this service.
- Table device in the core database. Add a new row for a test device, use the proper protocol_info_id. (Protocol_info_id is a foreign key to the protocol_info table in core).
- Table device_authorization. Add a new row to authorize owner for this device. (Function_group is a reference to the java enum DeviceFunctionGroup in platform/osgp-domain-core).

Changes to osgp-domain-newdomain

- Review entities. Be careful, the entities in this project are generated in the core database. The name of this project suggests that the entities would be generated in a domain specific database.
- Create valueobjects for your domain. The valueobjects in this project are used only in the adapter-ws and adapter-domain layer.

Changes to osgp-adapter-ws-newdomain

- Add Endpoints for each service request in presentation.ws.
- Add MessageProcessors in infra.jms.messageprocessors for each service response.
- Modify mapping/NewDomainMapper to map the JAXB generated classes to the classes in platform/osgp-domain-new-domain

Changes to osgp-adapter-domain-newdomain

- Add MessageProcessors in infra.jms.ws.messageprocessors for each service request.
- Add MessageProcessors in infra.jms.core.messageprocessors for each service response.
- Modify mapping/DomaiNewDomainMapper to map the classes in platform/osgp-domain-new-domain to the classes in shared/osgp-dto. The osgp-dto classes are used in the core layer and the protocol layer.

Testing the new domain services

In order to test the new domain services take a look at the Installation Guide. While following this guide keep the following items in mind:

Guidelines to add a new domain to GXF 219
A test device for the new domain must be available. This can either be a physical device or a simulated device.

- The test device must be connected or a device simulator must be running.
- The OSGP protocol adapter for the new device must be extended.
- ProgreSQL must be installed with all OSGP databases and system data as listed in the installation guide. The new domain might have a new database in which case the create script for the database and database owner must be run.
- Test data must be inserted into the following tables: organisation, device, device_authorization, device_function_mapping. Depending on the type of protocol adapter used for the new domain other tables might have to be populated as well. For instance a table like rtu_device for the IEC61850 Protocol Adapter.
- Apache Http Server must be installed and the new domain must be added to the configuration.
- Apache ActiveMQ must be installed.
- Tomcat application server must be installed and at least 4 web applications must be deployed:
  - An OSGP protocol adapter
  - OSGP Core
  - The OSGP Adapter Domain for your new domain
  - The OSGP Adapter WS for your new domain
- SoapUI can be used to test the new webservices for your domain.
Protocols

The open smart grid platform supported protocols can be found in this section. Feel free to add your own protocol or improve an existing protocol adapter.

Protocol Adapters

Protocol Adapter components translate a message from domain adapter components into a protocol message for a smart device. Protocol Adapter components send the protocol message to a smart device using a network connection. The response from the smart device is translated into a domain response message which will be sent to the Core components (which will route it to the domain adapter which issued the request).

OSLP

For the OSLP implementation, 2 components are used. The first component is the protocol adapter for the protocol. It can translate message into the protocol message for SSLD’s. Second there’s the signing-server component. It is responsible for signing the protocol message using the private key of the platform. The components communicate using a queue-pair. The signing-server can handle multiple protocol adapter instances by utilizing a reply-to queue per protocol adapter instance. Since the protocol adapter component needs to be reachable from a network, it is a requirement that the private key may not be used by the protocol adapter directly. The signing-server component can be deployed in such a way that no network access is available to this component, as the only coupling needed are the queues / the message broker.

DLMS

The DLMS protocol is used for (mainly) smart metering.

IEC61850

The IEC61850 implementation is used for e.g. distribution automation, microgrids and public lighting.

MQTT

The MQTT implementation is used in distribution automation

Basic layout of a protocol adapter

The following diagram shows the basic structure of a protocol adapter. This however does not mean that all protocol adapters will be structured like this, but instead the diagram shows an example of how such a protocol adapter could be structured.
IEC61850

IEC61850

The open smart grid platform supports IEC61850. IEC61850 is a popular protocol in the field of "smart grids". IEC61850 started as a standard for substation automation but has expanded into other domains such as EV and solar panels. Currently, the IEC61850 protocol is used within the Public Lighting, Microgrids and Distribution Automation domains. IEC61850 on Wikipedia

Protocol security

- No security options exist in this IEC61850 version 1 and 2
- Use through a secured tunnelling protocol like TLS (with client certificates) or VPN IEC Security guidelines can be found in IEC62351.

Specific communication service mapping (SCSM)

The open smart grid platform implementation supports:

- IEC 61850-8-1: Mappings to MMS (ISO/IEC9506-1 and ISO/IEC 9506-2)

Used library

The OpenMUC IEC61850 library from Fraunhofer is used to implement the protocol.

Supported Devices

These devices are currently supported by the Open Smart Grid Platform:

- Wago 750-881 RTU
- ABB 540CID11 RTU
- Kaifa AS101 load control box

Difference between OSLP and IEC61850

Contrary to OSLP the contract between OSGP and IEC61850 devices does not exist of request/response messages, instead the request messages received by OSGP will result in multiple read/write operations at the device. The response messages returned by OSGP will contain the result of these operations.
SWDevice-010805

Contract

Contract for SWDevice-010805. The contract specifies the messages which can be exchanged with an SSLD.

Messages

The messages below are part of OSGP and implemented in the IEC61850 protocol adapter and supported by the SWDevice-010805 device firmware.

- **RegisterDeviceRequest** (from device to platform) is a request that notifies the platform a device wants to register. During the registration the device sends its identification (serial number), and the device communicates its IP address to the platform.
- **RegisterDeviceResponse** (from platform to device) is a response which informs the device that the registration was successful. The device will not register anymore, until the next power cycle or reboot.
- **StartSelfTestRequest** (from platform to device) is a request which commands a device to switch all light relays on.
- **StartSelfTestResponse** (from device to platform) is a response which returns the result of the StartSelfTestRequest.
- **StopSelfTestRequest** (from platform to device) is a request which commands a device to switch all light relays off.
- **StopSelfTestResponse** (from device to platform) is a response which returns the result of the StopSelfTestRequest.
- **SetRebootRequest** (from platform to device) is a request which commands a device to reboot immediately.
- **SetRebootResponse** (from device to platform) is a response which returns the result of the SetRebootRequest.
- **SetLightRequest** (from platform to device) is a request which commands a device to switch on or off one ore several light relays.
- **SetLightResponse** (from device to platform) is a response which returns the result of the SetLightRequest.
- **SetTransitionRequest** (from platform to device) is a request which commands a device to switch its light relays according to light measurement schedule-entries.
- **SetTransitionResponse** (from device to platform) is a response which returns the result of the SetTransitionRequest.
- **SetEventNotificationsRequest** (from platform to device) is a request which commands a device to set the event notification mask.
- **SetEventNotificationsResponse** (from device to platform) is a response which returns the result of the SetEventNotificationsRequest.
- **GetStatusRequest** (from platform to device) is a request which queries a device for the current status of all its relays, the type of configuration (RELAY for SSLD), and the event notification mask set on the device.
- **GetStatusResponse** (from device to platform) is a response which returns the result of the GetStatusRequest and, if 'result = OK', returns the current status for all of the relays and other information.
- **GetFirmwareVersionRequest** (from platform to device) is a request which queries a device for its current firmware version.
- **GetFirmwareVersionResponse** (from device to platform) is a response which returns the result of the GetFirmwareVersionRequest and, if 'result = OK', returns the current firmware version.
- **UpdateFirmwareRequest** (from platform to device) is a request which commands a device to download a new firmware version from a server using a URL.
- **UpdateFirmwareResponse** (from device to platform) is a response which returns the result of the UpdateFirmwareRequest, which indicates if the device will start the process to download and install a new firmware version. Please note there are several events which are sent from the device to the platform to inform the platform when the firmware has been downloaded and whether or not the firmware was successfully activated.
- **GetConfigurationRequest** (from platform to device) is a request which queries a device for its current configuration settings.
- **GetConfigurationResponse** (from device to platform) is a response which returns the result of the GetConfigurationRequest and, if 'result = OK', returns the configuration settings retrieved from the device.
- **SetConfigurationRequest** (from platform to device) is a request which commands a device to update its configuration.

- **SetConfigurationResponse** (from device to platform) is a response which returns the result of the SetConfigurationRequest.

- **SetScheduleRequest** (from platform to device) is a request which commands a device to update its light or tariff schedule.

- **SetScheduleResponse** (from device to platform) is a response which returns the result of the SetScheduleRequest.

- **UpdateDeviceSslCertificationRequest** (from platform to device) is a request which commands a device to download a new certificate file from a server using a URL.

- **UpdateDeviceSslCertificationResponse** (from platform to device) is a response which returns the result of the UpdateDeviceSslCertificationRequest, which indicates if the device will start the process to download and install a new certificate file. Please note there are several events which are sent from the device to the platform to inform the platform whether or not the certificate file was successfully downloaded and activated.

- **EventNotificationRequest** (from device to platform) is a request that pushes an event notification from a device to the platform.

The following messages are not supported in IEC61850 and will return an UNSUPPORTED_DEVICE_ACTION SOAP Fault when a request is sent:

- ResumeScheduleRequest
- SwitchConfigurationRequest
- SwitchFirmwareRequest
- SetDeviceVerificationKeyRequest

The following message from device to OSGP is also not supported:

- ConfirmRegisterDeviceRequest
SSLD ICD file, SWDevice-010805.icd

<?xml version="1.0" encoding="UTF-8"?>
<!---Created by Test with OMICRON IEDScout 4.10 licensed to OMICRON electronics-->
<!--SCL Schema Version 3.1 (2012/10/22)-->
  <Communication>
    <SubNetwork name="NONE" type="8-MMS">
      <ConnectedAP apName="P1">
        <Address>
          <IP xsi:type="tp_IP">192.168.105.190</IP>
          <OSI-TSEL xsi:type="tP_OSI-TSEL">0001</OSI-TSEL>
          <OSI-SSEL xsi:type="tP_OSI-SSEL">0001</OSI-SSEL>
          <OSI-AP-Title>1,1,1,999,1</OSI-AP-Title>
          <OSI-AP-Invoke xsi:type="tP_OSI-AP-Invoke">0</OSI-AP-Invoke>
          <OSI-AE-Qualifier xsi:type="tP_OSI-AE-Qualifier">12</OSI-AE-Qualifier>
          <OSI-AE-Invoke xsi:type="tP_OSI-AE-Invoke">0</OSI-AE-Invoke>
          <MMS-Port xsi:type="tP_MMS-Port">192</MMS-Port>
        </Address>
      </ConnectedAP>
    </SubNetwork>
    <IED name="SWDevice">
      <Services>
        <DynAssociation />
        <GetDirectory />
        <GetDataObjectDefinition />
        <DataSetDirection />
        <GetDataSetValue />
        <SetDataSetValue />
        <DataSetDirectory />
        <ConfDataSet max="1" modify="false" />
        <DynDataSet max="42" />
        <ReadWrite />
        <ConfReportControl max="1" bufConf="false" />
        <GetCBValues />
        <ReportSettings rptID="Dyn" optFields="Dyn" bufTime="Dyn" trgOps="Dyn" intgPd="Dyn"
          fixPrefix="true" fixLnInst="true" />
        <GOOSE max="0" />
        <GSSE max="0" />
      </Services>
    </IED>
    <AccessPoint name="P1">
      <Server>
        <Authentication none="true" />
        <Device inst="GenericIO">
          <LN0 lnType="SWDeviceGenericIO.LLN0" lnClass="LLN0" inst=""/>
          <DataSet name="evn_rpn">
            <FCDA ldInst="GenericIO" prefix="" lnClass="CSLC" lnInst="" doName=""/>
          </DataSet>
          <ReportControl name="evn_rpn01" rptID="evn_rpn" buffered="true" buf=False
            seqNum="true" qchg="true" chg="false" dupd="false" period="false" gi="false"
            OptFields seqNum="true" timeFsm="true" dataSet="true" reasonX="false"
            RptEnabled max="1" />
        </ReportControl>
        <DOI name="Mod">
          <DAI name="ctlModel">
            <Val>status-only</Val>
          </DAI>
          </DOI>
        <DOI name="NamPlt">
          <DAI name="vendor">
            <Val>MZ Automation</Val>
          </DAI>
          </DOI>
        <DOI name="swRev">
          <DAI name="P1">
            <Val>0.0.1</Val>
          </DAI>
        </DOI>
      </Server>
    </AccessPoint>
  </Communication>
</SCL>
<Val>libiec61850 server example</Val>
</DAI>
<DAI name="configRev">
<Val></Val>
</DAI>
<DAI name="ldNs">
<Val></Val>
</DAI>
</DOI>
</LN>
<LN lnType="SWDeviceGenericIO.CSLC" lnClass="CSLC" inst="">
<DOI name="RbOper">
<DAI name="ctlModel">
<Val>status-only</Val>
</DAI>
</DOI>
<DOI name="TlsCf">
<DAI name="enbTls">
<Val></Val>
</DAI>
<DAI name="port">
<Val></Val>
</DAI>
<DAI name="comName">
<Val></Val>
</DAI>
</DOI>
<DOI name="Sensor">
<DAI name="ctlModel">
<Val>status-only</Val>
</DAI>
</DOI>
<DOI name="Attnm">
<DAI name="lon">
<Val>5.591</Val>
</DAI>
<DAI name="lat">
<Val>52.143</Val>
</DAI>
<DAI name="zenith">
<Val></Val>
</DAI>
</DOI>
<DOI name="SWCf">
<DAI name="LT">
<Val>RELAY</Val>
</DAI>
<DAI name="adSetOft">
<Val></Val>
</DAI>
<DAI name="adRiseOft">
<Val></Val>
</DAI>
</DOI>
<DOI name="FuncFwDw">
<DAI name="url">
<Val></Val>
</DAI>
</DOI>
<DOI name="Clock">
<DAI name="tZ">
<Val>60</Val>
</DAI>
<DAI name="dstBegT">
<Val>M3.5.0/2</Val>
</DAI>
<DAI name="dstEndT">
<Val>M10.5.0/3</Val>
</DAI>
<DAI name="dvt">
<Val>60</Val>
</DAI>
<DAI name="enbDst">  
  <Val>1</Val>  
</DAI>
<DAI name="enbNtpC">  
  <Val>1</Val>  
</DAI>
<DAI name="ntpSvrA">  
  <Val>192.168.105.200</Val>  
</DAI>
<DAI name="syncPer">  
  <Val>1440</Val>  
</DAI>
<DAI name="minOst">  
  <Val>2</Val>  
</DAI>
<DAI name="almOst">  
  <Val>5</Val>  
</DAI>
<DAI name="maxOst">  
  <Val>900</Val>  
</DAI>
</DOI>
<DOI name="Reg">  
  <DAI name="svrAddr">  
    <Val>10.2.1.127</Val>  
  </DAI>  
  <DAI name="svrPort">  
    <Val>50000</Val>  
  </DAI>  
  <DAI name="ntfEnb">  
    <Val>1</Val>  
  </DAI>  
</DOI>
<DOI name="Log">  
  <DAI name="enb">  
    <Val>1</Val>  
  </DAI>  
</DOI>
<DOI name="ScyFwDw">  
  <DAI name="url">  
    <Val></Val>  
  </DAI>  
</DOI>
<DOI name="CARepl">  
  <DAI name="url">  
    <Val></Val>  
  </DAI>  
</DOI>
<DOI name="IPCf">  
  <DAI name="enbDHCP">  
    <Val>1</Val>  
  </DAI>  
  <DAI name="ipAddr">  
    <Val>192.168.105.227</Val>  
  </DAI>  
  <DAI name="netmask">  
    <Val>255.255.255.0</Val>  
  </DAI>  
  <DAI name="gateway">  
    <Val>192.168.1.1</Val>  
  </DAI>  
</DOI>
<DOI name="EvnBuf">  
  <DAI name="enbEvnType">  
    <Val>1FFFFFF</Val>  
  </DAI>  
</DOI>
<DOI name="SNMP Cf">  
  <DAI name="userName">  
    <Val>sw_device</Val>  
  </DAI>  
</DOI>
<DAI name="tOffT">
  <Val>0</Val>
</DAI>
<DAI name="minOnPer">
  <Val>30</Val>
</DAI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche4">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche5">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
</SDI>
<DAI name="tOn">
  <Val>0</Val>
</DAI>
<DAI name="tOnT">
  <Val>0</Val>
</DAI>
<DAI name="tOff">
  <Val>0</Val>
</DAI>
<DAI name="tOffT">
  <Val>0</Val>
</DAI>
<DAI name="minOnPer">
  <Val>30</Val>
</DAI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche6">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
</SDI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche7">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
</SDI>
<SDI name="sche8">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
</SDI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche9">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche10">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
  <Val>30</Val>
</DAI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche11">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="toffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche12">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="toffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<DAI name="tOn">  
<Val>0</Val>  
</DAI>  
<DAI name="tOnT">  
<Val>0</Val>  
</DAI>  
<DAI name="tOff">  
<Val>0</Val>  
</DAI>  
<DAI name="tOffT">  
<Val>0</Val>  
</DAI>  
<DAI name="minOnPer">  
<Val>30</Val>  
</DAI>  
<DAI name="minOffPer">  
<Val>30</Val>  
</DAI>  
<DAI name="srBefWd">  
<Val>30</Val>  
</DAI>  
<DAI name="srAftWd">  
<Val>30</Val>  
</DAI>  
<DAI name="igBefWd">  
<Val>0</Val>  
</DAI>  
<DAI name="igAftWd">  
<Val>0</Val>  
</DAI>  
<DAI name="Descr">  
<Val>ON/OFF schedule</Val>  
</DAI>  
</SDI>  
<SDI name="sche13">  
<DAI name="enable">  
<Val>0</Val>  
</DAI>  
<DAI name="day">  
<Val>0</Val>  
</DAI>  
<DAI name="tOn">  
<Val>0</Val>  
</DAI>  
<DAI name="tOnT">  
<Val>0</Val>  
</DAI>  
<DAI name="tOff">  
<Val>0</Val>  
</DAI>  
<DAI name="tOffT">  
<Val>0</Val>  
</DAI>  
<DAI name="minOnPer">  
<Val>30</Val>  
</DAI>  
<DAI name="minOffPer">  
<Val>30</Val>  
</DAI>  
<DAI name="srBefWd">  
<Val>30</Val>  
</DAI>  
<DAI name="srAftWd">  
<Val>30</Val>  
</DAI>  
<DAI name="igBefWd">  
<Val>0</Val>  
</DAI>  
<DAI name="igAftWd">  
<Val>0</Val>  
</DAI>  
<DAI name="Descr">  
<Val>ON/OFF schedule</Val>  
</DAI>  
</SDI>
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche14">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>

<SDI name="sche15">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
</SDI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche16">
<DAI name="enable">
<Val>0</Val>
</DAI>
<DAI name="day">
<Val>0</Val>
</DAI>
<DAI name="tOn">
<Val>0</Val>
</DAI>
<DAI name="tOnT">
<Val>0</Val>
</DAI>
<DAI name="tOff">
<Val>0</Val>
</DAI>
<DAI name="tOffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
</DAI>
<DAI name="minOnPer">
  <Val>30</Val>
</DAI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche18">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="toffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche19">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn"
ON/OFF schedule
<DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</DAI>

</SDI>
<SDI name="sche27">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI>
<SDI name="sche28">
 <DAI name="enable">
  <Val>0</Val>
 </DAI>
 <DAI name="day">
  <Val>0</Val>
 </DAI>
 <DAI name="tOn">
  <Val>0</Val>
 </DAI>
 <DAI name="tOnT">
  <Val>0</Val>
 </DAI>
 <DAI name="toff">
  <Val>0</Val>
 </DAI>
 <DAI name="tOff">
  <Val>0</Val>
 </DAI>
 <DAI name="tOffT">
  <Val>0</Val>
 </DAI>
 <DAI name="minOnPer">
  <Val>30</Val>
 </DAI>
 <DAI name="minOffPer">
  <Val>30</Val>
 </DAI>
 <DAI name="srBefWd">
  <Val>30</Val>
 </DAI>
 <DAI name="srAftWd">
  <Val>30</Val>
 </DAI>
 <DAI name="igBefWd">
  <Val>0</Val>
 </DAI>
 <DAI name="igAftWd">
  <Val>0</Val>
 </DAI>
 <DAI name="Descr">
  <Val>ON/OFF schedule</Val>
 </DAI>
</SDI>
</SDI>
<SDI name="sche29">
 <DAI name="enable">
  <Val>0</Val>
 </DAI>
 <DAI name="day">
  <Val>0</Val>
 </DAI>
 <DAI name="tOn">
  <Val>0</Val>
 </DAI>
 <DAI name="tOnT">
  <Val>0</Val>
 </DAI>
 <DAI name="toff">
  <Val>0</Val>
 </DAI>
 <DAI name="tOff">
  <Val>0</Val>
 </DAI>
 <DAI name="tOffT">
  <Val>0</Val>
 </DAI>
 <DAI name="minOnPer">
  <Val>30</Val>
 </DAI>
 <DAI name="minOffPer">
  <Val>30</Val>
 </DAI>
 <DAI name="srBefWd">
  <Val>30</Val>
 </DAI>
 <DAI name="srAftWd">
  <Val>30</Val>
 </DAI>
 <DAI name="igBefWd">
  <Val>0</Val>
 </DAI>
 <DAI name="igAftWd">
  <Val>0</Val>
 </DAI>
 <DAI name="Descr">
  <Val>ON/OFF schedule</Val>
 </DAI>
</SDI>
</SDI>
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche32">
<DAI name="enable">
<Val>0</Val>
</DAI>
<DAI name="day">
<Val>0</Val>
</DAI>
<DAI name="tOn">
<Val>0</Val>
</DAI>
<DAI name="tOnT">
<Val>0</Val>
</DAI>
<DAI name="tOff">
<Val>0</Val>
</DAI>
<DAI name="tOffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche33">
<DAI name="enable">
<Val>0</Val>
</DAI>
<DAI name="day">
<Val>0</Val>
</DAI>
<DAI name="tOn">
<Val>0</Val>
</DAI>
<DAI name="tOnT">
  <Val>0</Val>
</DAI>
<DAI name="tOffT">
  <Val>0</Val>
</DAI>
<DAI name="minOnPer">
  <Val>30</Val>
</DAI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche34">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche35">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>

<SDI name="sche36">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>

<SDI name="sche37">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>

<SDI name="sche38">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
</SDI>
<DAI name="minOffPer"/>
  <Val>30</Val>
</DAI>
<DAI name="srBefWd"/>
  <Val>30</Val>
</DAI>
<DAI name="srAftWd"/>
  <Val>30</Val>
</DAI>
<DAI name="igBefWd"/>
  <Val>0</Val>
</DAI>
<DAI name="igAftWd"/>
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche39">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
</SDI>
<SDI name="sche40">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<Val>0</Val>
</DAI>
<DAI name="tOff">
<Val>0</Val>
</DAI>
<DAI name="tOffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche41">
<DAI name="enable">
<Val>0</Val>
</DAI>
<DAI name="day">
<Val>0</Val>
</DAI>
<DAI name="tOn">
<Val>0</Val>
</DAI>
<DAI name="tOnT">
<Val>0</Val>
</DAI>
<DAI name="tOff">
<Val>0</Val>
</DAI>
<DAI name="tOffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche42"/>
<DAI name="enable">
  <Val>0</Val>
</DAI>
<DAI name="day">
  <Val>0</Val>
</DAI>
<DAI name="tOn">
  <Val>0</Val>
</DAI>
<DAI name="tOnT">
  <Val>0</Val>
</DAI>
<DAI name="tOff">
  <Val>0</Val>
</DAI>
<DAI name="tOffT">
  <Val>0</Val>
</DAI>
<DAI name="minOnPer">
  <Val>30</Val>
</DAI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche43">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
ON/OFF schedule

SDI name="sche44"
</SDI>

SDI name="sche45"
</SDI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche46">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche47">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
</SDI>

<DAI name="tOff">
  <Val>0</Val>
</DAI>
<DAI name="tOffT">
  <Val>0</Val>
</DAI>
<DAI name="minOnPer">
  <Val>30</Val>
</DAI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche48">
<DAI name="enable">
  <Val>0</Val>
</DAI>
<DAI name="day">
  <Val>0</Val>
</DAI>
<DAI name="tOn">
  <Val>0</Val>
</DAI>
<DAI name="tOnT">
  <Val>0</Val>
</DAI>
<DAI name="tOff">
  <Val>0</Val>
</DAI>
<DAI name="tOffT">
  <Val>0</Val>
</DAI>
<DAI name="minOnPer">
  <Val>30</Val>
</DAI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche49">
<DAI name="enable">
  <Val>0</Val>
</DAI>
</SDI>
<Val>0</Val>
</DAI>
<DAI name="day">
 <Val>0</Val>
</DAI>
<DAI name="tOn">
 <Val>0</Val>
</DAI>
<DAI name="tOnT">
 <Val>0</Val>
</DAI>
<DAI name="tOff">
 <Val>0</Val>
</DAI>
<DAI name="tOffT">
 <Val>0</Val>
</DAI>
<DAI name="minOnPer">
 <Val>30</Val>
</DAI>
<DAI name="minOffPer">
 <Val>30</Val>
</DAI>
<DAI name="srBefWd">
 <Val>30</Val>
</DAI>
<DAI name="srAftWd">
 <Val>30</Val>
</DAI>
<DAI name="igBefWd">
 <Val>0</Val>
</DAI>
<DAI name="igAftWd">
 <Val>0</Val>
</DAI>
<DAI name="Descr">
 <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche50">
 <DAI name="enable">
  <Val>0</Val>
 </DAI>
 <DAI name="day">
  <Val>0</Val>
 </DAI>
 <DAI name="tOn">
  <Val>0</Val>
 </DAI>
 <DAI name="tOnT">
  <Val>0</Val>
 </DAI>
 <DAI name="tOff">
  <Val>0</Val>
 </DAI>
 <DAI name="tOffT">
  <Val>0</Val>
 </DAI>
 <DAI name="minOnPer">
  <Val>30</Val>
 </DAI>
 <DAI name="minOffPer">
  <Val>30</Val>
 </DAI>
 <DAI name="srBefWd">
  <Val>30</Val>
 </DAI>
 <DAI name="srAftWd">
  <Val>30</Val>
 </DAI>
 <DAI name="igBefWd">
  <Val>0</Val>
 </DAI>
 <DAI name="igAftWd">
  <Val>0</Val>
 </DAI>
 <DAI name="Descr">
  <Val>ON/OFF schedule</Val>
 </DAI>
</SDI>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche51">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche52">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
</SDI>
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche53">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche54">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
<DAI name="day">
  <Val>0</Val>
</DAI>
<DAI name="tOn">
  <Val>0</Val>
</DAI>
<DAI name="tOnT">
  <Val>0</Val>
</DAI>
<DAI name="tOff">
  <Val>0</Val>
</DAI>
<DAI name="tOffT">
  <Val>0</Val>
</DAI>
<DAI name="minOnPer">
  <Val>30</Val>
</DAI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SI>

<SI name="sche57">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche58">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche59">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
</SDI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche60">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche61">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
</SDI>
<DAI name="day">
  <Val>0</Val>
</DAI>
<DAI name="tOn">
  <Val>0</Val>
</DAI>
<DAI name="tOnT">
  <Val>0</Val>
</DAI>
<DAI name="tOff">
  <Val>0</Val>
</DAI>
<DAI name="tOffT">
  <Val>0</Val>
</DAI>
<DAI name="minOnPer">
  <Val>30</Val>
</DAI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche64">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>

<SDI name="sche1">
  <DAI name="enable">
    <Val>1</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>1</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>2</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>

<SDI name="sche2">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
</SDI>

<LN lnType="SWDeviceGenericIO.XSWC1" lnClass="XSWC" inst="2">
  <DOI name="SwType">
    <DAI name="ctlModel">
      <Val>status-only</Val>
    </DAI>
  </DOI>
  <DOI name="Pos">
    <DAI name="ctlModel">
      <Val>status-only</Val>
    </DAI>
  </DOI>
  <DOI name="CfSt">
    <DAI name="enbOper">
      <Val>1</Val>
    </DAI>
  </DOI>
  <DOI name="Sche">
    <SDI name="sche1">
      <DAI name="enable">
        <Val>1</Val>
      </DAI>
      <DAI name="day">
        <Val>0</Val>
      </DAI>
      <DAI name="tOn">
        <Val>0</Val>
      </DAI>
      <DAI name="tOnT">
        <Val>1</Val>
      </DAI>
      <DAI name="tOff">
        <Val>0</Val>
      </DAI>
      <DAI name="tOffT">
        <Val>2</Val>
      </DAI>
      <DAI name="minOnPer">
        <Val>30</Val>
      </DAI>
      <DAI name="minOffPer">
        <Val>30</Val>
      </DAI>
      <DAI name="srBefWd">
        <Val>30</Val>
      </DAI>
      <DAI name="srAftWd">
        <Val>30</Val>
      </DAI>
      <DAI name="igBefWd">
        <Val>0</Val>
      </DAI>
      <DAI name="igAftWd">
        <Val>0</Val>
      </DAI>
      <DAI name="Descr">
        <Val>ON/OFF schedule</Val>
      </DAI>
    </SDI>
    <SDI name="sche2">
      <DAI name="enable">
        <Val>0</Val>
      </DAI>
      <DAI name="day">
        <Val>0</Val>
      </DAI>
    </SDI>
  </DOI>
</LN>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche4">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche5">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
</SDI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
<SDI name="sche6">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche7">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<DAI name="minOnPer">30</DAI>
<DAI name="minOffPer">30</DAI>
<DAI name="srBefWd">30</DAI>
<DAI name="srAftWd">30</DAI>
<DAI name="igBefWd">0</DAI>
<DAI name="igAftWd">0</DAI>
<DAI name="Descr">ON/OFF schedule</DAI>
<SDI name="sche8">
<DAI name="enable">0</DAI>
<DAI name="day">0</DAI>
<DAI name="tOn">0</DAI>
<DAI name="tOnT">0</DAI>
<DAI name="tOff">0</DAI>
<DAI name="tOffT">0</DAI>
<DAI name="minOnPer">30</DAI>
<DAI name="minOffPer">30</DAI>
<DAI name="srBefWd">30</DAI>
<DAI name="srAftWd">30</DAI>
<DAI name="igBefWd">0</DAI>
<DAI name="igAftWd">0</DAI>
<DAI name="Descr">ON/OFF schedule</DAI>
</SDI>
<SDI name="sche9">
<DAI name="enable">0</DAI>
<DAI name="day">0</DAI>
<DAI name="tOn">0</DAI>
<DAI name="tOnT">0</DAI>
<DAI name="tOff">0</DAI>
<DAI name="tOffT">0</DAI>
<DAI name="minOnPer">30</DAI>
<DAI name="minOffPer">30</DAI>
<DAI name="srBefWd">30</DAI>
<DAI name="srAftWd">30</DAI>
<DAI name="igBefWd">0</DAI>
<DAI name="igAftWd">0</DAI>
<DAI name="Descr">ON/OFF schedule</DAI>
</SDI>
<DAI name="tOn">  <Val>0</Val> </DAI>  
<DAI name="tOnT">  <Val>0</Val> </DAI>  
<DAI name="tOff">  <Val>0</Val> </DAI>  
<DAI name="tOffT">  <Val>0</Val> </DAI>  
<DAI name="minOnPer">  <Val>30</Val> </DAI>  
<DAI name="minOffPer">  <Val>30</Val> </DAI>  
<DAI name="srBefWd">  <Val>30</Val> </DAI>  
<DAI name="srAftWd">  <Val>30</Val> </DAI>  
<DAI name="igBefWd">  <Val>0</Val> </DAI>  
<DAI name="igAftWd">  <Val>0</Val> </DAI>  
<DAI name="Descr">  <Val>ON/OFF schedule</Val> </DAI>  
</SDI>  
<SDI name="sche10">  
<DAI name="enable">  <Val>0</Val> </DAI>  
<DAI name="day">  <Val>0</Val> </DAI>  
<DAI name="tOn">  <Val>0</Val> </DAI>  
<DAI name="tOnT">  <Val>0</Val> </DAI>  
<DAI name="tOff">  <Val>0</Val> </DAI>  
<DAI name="tOffT">  <Val>0</Val> </DAI>  
<DAI name="minOnPer">  <Val>30</Val> </DAI>  
<DAI name="minOffPer">  <Val>30</Val> </DAI>  
<DAI name="srBefWd">  <Val>30</Val> </DAI>  
<DAI name="srAftWd">  <Val>30</Val> </DAI>  
<DAI name="igBefWd">  <Val>0</Val> </DAI>  
<DAI name="igAftWd">  <Val>0</Val> </DAI>  
<DAI name="Descr"> </DAI>  
</SDI>
ON/OFF schedule

DAI

SDI name="sche11">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>

SDI name="sche12">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
</SDI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche13">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
</SDI>
<SDI name="sche14">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<DAI name="minOnPer">
  <Val>30</Val>
</DAI>

<DAI name="minOffPer">
  <Val>30</Val>
</DAI>

<DAI name="srBefWd">
  <Val>30</Val>
</DAI>

<DAI name="srAftWd">
  <Val>30</Val>
</DAI>

<DAI name="igBefWd">
  <Val>0</Val>
</DAI>

<DAI name="igAftWd">
  <Val>0</Val>
</DAI>

<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>

</SDI>

<SDI name="sche15">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>

<SDI name="sche16">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
ON/OFF schedule
ON/OFF schedule

<DAI name="sche18">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</DAI>

<SDI name="sche19">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
</SDI>
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche20">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="toffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche21">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="toffT">
    <Val>0</Val>
  </DAI>
</SDI>
<DAI name="minOnPer">
  <Val>30</Val>
</DAI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche22">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche23">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
<DAI name="tOnT">
  <Val>0</Val>
</DAI>
<DAI name="tOff">
  <Val>0</Val>
</DAI>
<DAI name="tOffT">
  <Val>0</Val>
</DAI>
<DAI name="minOnPer">
  <Val>30</Val>
</DAI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche24">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
</DAI>
<SDI name="sche25">
    <DAI name="enable">
        <Val>0</Val>
    </DAI>
    <DAI name="day">
        <Val>0</Val>
    </DAI>
    <DAI name="tOn">
        <Val>0</Val>
    </DAI>
    <DAI name="tOnT">
        <Val>0</Val>
    </DAI>
    <DAI name="tOff">
        <Val>0</Val>
    </DAI>
    <DAI name="tOffT">
        <Val>0</Val>
    </DAI>
    <DAI name="minOnPer">
        <Val>30</Val>
    </DAI>
    <DAI name="minOffPer">
        <Val>30</Val>
    </DAI>
    <DAI name="srBefWd">
        <Val>30</Val>
    </DAI>
    <DAI name="srAftWd">
        <Val>30</Val>
    </DAI>
    <DAI name="igBefWd">
        <Val>0</Val>
    </DAI>
    <DAI name="igAftWd">
        <Val>0</Val>
    </DAI>
    <DAI name="Descr">
        <Val>ON/OFF schedule</Val>
    </DAI>
</SDI>

<SDI name="sche26">
    <DAI name="enable">
        <Val>0</Val>
    </DAI>
    <DAI name="day">
        <Val>0</Val>
    </DAI>
    <DAI name="tOn">
        <Val>0</Val>
    </DAI>
    <DAI name="tOnT">
        <Val>0</Val>
    </DAI>
    <DAI name="tOff">
        <Val>0</Val>
    </DAI>
    <DAI name="tOffT">
        <Val>0</Val>
    </DAI>
    <DAI name="minOnPer">
        <Val>30</Val>
    </DAI>
    <DAI name="minOffPer">
        <Val>30</Val>
    </DAI>
    <DAI name="srBefWd">
        <Val>30</Val>
    </DAI>
    <DAI name="srAftWd">
        <Val>30</Val>
    </DAI>
</SDI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche27">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche28">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
</SDI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche29">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche30">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
</DAI>
<DAI name="tOnT">  
<Val>0</Val>  
</DAI>  
<DAI name="tOff">  
<Val>0</Val>  
</DAI>  
<DAI name="tOffT">  
<Val>0</Val>  
</DAI>  
<DAI name="minOnPer">  
<Val>30</Val>  
</DAI>  
<DAI name="minOffPer">  
<Val>30</Val>  
</DAI>  
<DAI name="srBefWd">  
<Val>30</Val>  
</DAI>  
<DAI name="srAftWd">  
<Val>30</Val>  
</DAI>  
<DAI name="igBefWd">  
<Val>0</Val>  
</DAI>  
<DAI name="igAftWd">  
<Val>0</Val>  
</DAI>  
<DAI name="Descr">  
<Val>ON/OFF schedule</Val>  
</DAI>  
</SDI>  
<SDI name="sche31">  
<DAI name="enable">  
<Val>0</Val>  
</DAI>  
<DAI name="day">  
<Val>0</Val>  
</DAI>  
<DAI name="tOn">  
<Val>0</Val>  
</DAI>  
<DAI name="tOnT">  
<Val>0</Val>  
</DAI>  
<DAI name="tOff">  
<Val>0</Val>  
</DAI>  
<DAI name="tOffT">  
<Val>0</Val>  
</DAI>  
<DAI name="minOnPer">  
<Val>30</Val>  
</DAI>  
<DAI name="minOffPer">  
<Val>30</Val>  
</DAI>  
<DAI name="srBefWd">  
<Val>30</Val>  
</DAI>  
<DAI name="srAftWd">  
<Val>30</Val>  
</DAI>  
<DAI name="igBefWd">  
<Val>0</Val>  
</DAI>  
<DAI name="igAftWd">  
<Val>0</Val>  
</DAI>  
<DAI name="Descr">  
<Val>ON/OFF schedule</Val>  
</DAI>  
</SDI>
<SDI name="sche32">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>

<SDI name="sche33">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
<DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche34">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
</SDI>
<SDI name="sche35">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
</SDI>
<Val>0</Val>
</DAI>
<DAI name="tOff">
<Val>0</Val>
</DAI>
<DAI name="tOffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche38">
<DAI name="enable">
<Val>0</Val>
</DAI>
<DAI name="day">
<Val>0</Val>
</DAI>
<DAI name="tOn">
<Val>0</Val>
</DAI>
<DAI name="tOnT">
<Val>0</Val>
</DAI>
<DAI name="tOff">
<Val>0</Val>
</DAI>
<DAI name="tOffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<DAI name="enable">
  <Val>0</Val>
</DAI>
<DAI name="day">
  <Val>0</Val>
</DAI>
<DAI name="tOn">
  <Val>0</Val>
</DAI>
<DAI name="tOnT">
  <Val>0</Val>
</DAI>
<DAI name="tOff">
  <Val>0</Val>
</DAI>
<DAI name="tOffT">
  <Val>0</Val>
</DAI>
<DAI name="minOnPer">
  <Val>30</Val>
</DAI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche40">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche41">
<DAI name="enable">
<Val>0</Val>
</DAI>
<DAI name="day">
<Val>0</Val>
</DAI>
<DAI name="tOn">
<Val>0</Val>
</DAI>
<DAI name="tOnT">
<Val>0</Val>
</DAI>
<DAI name="tOff">
<Val>0</Val>
</DAI>
<DAI name="tOffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche42">
<DAI name="enable">
<Val>0</Val>
</DAI>
<DAI name="day">
<Val>0</Val>
</DAI>
<DAI name="tOn">
<Val>0</Val>
</DAI>
<DAI name="tOnT">
<Val>0</Val>
</DAI>
<DAI name="tOff">
<Val>0</Val>
</DAI>
<DAI name="tOffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche43">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche44">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</DAI>

<SDI name="sche45">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>

<SDI name="sche46">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<Val>0</Val>
</DAI>
<DAI name="day">
  <Val>0</Val>
</DAI>
<DAI name="tOn">
  <Val>0</Val>
</DAI>
<DAI name="tOnT">
  <Val>0</Val>
</DAI>
<DAI name="tOff">
  <Val>0</Val>
</DAI>
<DAI name="tOffT">
  <Val>0</Val>
</DAI>
<DAI name="minOnPer">
  <Val>30</Val>
</DAI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche47">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>

<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>

<SDI name="sche48">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>

<SDI name="sche49">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
</SDI>
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche50">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche51">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<DAI name="tOff">
  <Val>0</Val>
</DAI>
<DAI name="tOffT">
  <Val>0</Val>
</DAI>
<DAI name="minOnPer">
  <Val>30</Val>
</DAI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche52">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche53">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
</SDI>
ON/OFF schedule

<Val>8</Val>
</DAI>
<DAI name="tOffT">
<Val>8</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>8</Val>
</DAI>
<DAI name="igAftWd">
<Val>8</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche59">
<DAI name="enable">
<Val>8</Val>
</DAI>
<DAI name="day">
<Val>8</Val>
</DAI>
<DAI name="tOn">
<Val>8</Val>
</DAI>
<DAI name="tOnT">
<Val>8</Val>
</DAI>
<DAI name="tOff">
<Val>8</Val>
</DAI>
<DAI name="tOffT">
<Val>8</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>8</Val>
</DAI>
<DAI name="igAftWd">
<Val>8</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche60">
<DAI name="enable">
<Val>8</Val>
</DAI>

<DAI name="day">
  <Val>0</Val>
</DAI>
<DAI name="tOn">
  <Val>0</Val>
</DAI>
<DAI name="tOnT">
  <Val>0</Val>
</DAI>
<DAI name="tOff">
  <Val>0</Val>
</DAI>
<DAI name="tOffT">
  <Val>0</Val>
</DAI>
<DAI name="minOnPer">
  <Val>30</Val>
</DAI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche61">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
ON/OFF schedule

<DAI name="enable">
  <Val>0</Val>
</DAI>

<DAI name="day">
  <Val>0</Val>
</DAI>

<DAI name="tOn">
  <Val>0</Val>
</DAI>

<DAI name="tOnT">
  <Val>0</Val>
</DAI>

<DAI name="tOff">
  <Val>0</Val>
</DAI>

<DAI name="tOffT">
  <Val>0</Val>
</DAI>

<DAI name="minOnPer">
  <Val>30</Val>
</DAI>

<DAI name="minOffPer">
  <Val>30</Val>
</DAI>

<DAI name="srBefWd">
  <Val>30</Val>
</DAI>

<DAI name="srAftWd">
  <Val>30</Val>
</DAI>

<DAI name="igBefWd">
  <Val>0</Val>
</DAI>

<DAI name="igAftWd">
  <Val>0</Val>
</DAI>

<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche64">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
</DOI>
</LN>
<LN lnType="SWDeviceGenericIO.XSWC1" lnClass="XSWC" inst="3">
  <DOI name="SwType">
    <DAI name="ctlModel">
      <Val>status-only</Val>
    </DAI>
  </DOI>
  <DOI name="Pos">
    <DAI name="ctlModel">
      <Val>status-only</Val>
    </DAI>
  </DOI>
  <DOI name="CfSt">
    <DAI name="enbOper">
      <Val></Val>
    </DAI>
  </DOI>
</LN>
<SDI name="sche1">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>

<SDI name="sche2">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
</SDI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche3">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche4">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
</SDI>
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
  <Val>30</Val>
</DAI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche5">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="toffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche6">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
</SDI>
<DAI name="tOn">
<Val>0</Val>
</DAI>
<DAI name="tOnT">
<Val>0</Val>
</DAI>
<DAI name="tOff">
<Val>0</Val>
</DAI>
<DAI name="tOffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche7">
<DAI name="enable">
<Val>0</Val>
</DAI>
<DAI name="day">
<Val>0</Val>
</DAI>
<DAI name="tOn">
<Val>0</Val>
</DAI>
<DAI name="tOnT">
<Val>0</Val>
</DAI>
<DAI name="tOff">
<Val>0</Val>
</DAI>
<DAI name="tOffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche8">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>

<SDI name="sche9">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
</SDI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>

<SDI name="sche10">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>

<SDI name="sche11">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
</SDI>
<DAI name="minOnPer">
    <Val>30</Val>
</DAI>

<DAI name="minOffPer">
    <Val>30</Val>
</DAI>

<DAI name="srBefWd">
    <Val>30</Val>
</DAI>

<DAI name="srAftWd">
    <Val>30</Val>
</DAI>

<DAI name="igBefWd">
    <Val>0</Val>
</DAI>

<DAI name="igAftWd">
    <Val>0</Val>
</DAI>

<DAI name="Descr">
    <Val>ON/OFF schedule</Val>
</DAI>

<SDI name="sche12">
    <DAI name="enable">
        <Val>0</Val>
    </DAI>
    <DAI name="day">
        <Val>0</Val>
    </DAI>
    <DAI name="tOn">
        <Val>0</Val>
    </DAI>
    <DAI name="tOnT">
        <Val>0</Val>
    </DAI>
    <DAI name="tOff">
        <Val>0</Val>
    </DAI>
    <DAI name="tOffT">
        <Val>0</Val>
    </DAI>
    <DAI name="minOnPer">
        <Val>30</Val>
    </DAI>
    <DAI name="minOffPer">
        <Val>30</Val>
    </DAI>
    <DAI name="srBefWd">
        <Val>30</Val>
    </DAI>
    <DAI name="srAftWd">
        <Val>30</Val>
    </DAI>
    <DAI name="igBefWd">
        <Val>0</Val>
    </DAI>
    <DAI name="igAftWd">
        <Val>0</Val>
    </DAI>
    <DAI name="Descr">
        <Val>ON/OFF schedule</Val>
    </DAI>
</SDI>

<SDI name="sche13">
    <DAI name="enable">
        <Val>0</Val>
    </DAI>
    <DAI name="day">
        <Val>0</Val>
    </DAI>
    <DAI name="tOn">
<Val>0</Val>
</DAI>
<DAI name="t0nT">
<Val>0</Val>
</DAI>
<DAI name="t0ff">
<Val>0</Val>
</DAI>
<DAI name="t0ffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche14">
<DAI name="enable">
<Val>0</Val>
</DAI>
<DAI name="day">
<Val>0</Val>
</DAI>
<DAI name="t0n">
<Val>0</Val>
</DAI>
<DAI name="t0nT">
<Val>0</Val>
</DAI>
<DAI name="t0ff">
<Val>0</Val>
</DAI>
<DAI name="t0ffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<DAI>
</SDI>
<SDI name="sche15"/>
<DAI name="enable">
  <Val>0</Val>
</DAI>
<DAI name="day">
  <Val>0</Val>
</DAI>
<DAI name="tOn">
  <Val>0</Val>
</DAI>
<DAI name="tOnT">
  <Val>0</Val>
</DAI>
<DAI name="tOff">
  <Val>0</Val>
</DAI>
<DAI name="tOffT">
  <Val>0</Val>
</DAI>
<DAI name="minOnPer">
  <Val>30</Val>
</DAI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche16"/>
<DAI name="enable">
  <Val>0</Val>
</DAI>
<DAI name="day">
  <Val>0</Val>
</DAI>
<DAI name="tOn">
  <Val>0</Val>
</DAI>
<DAI name="tOnT">
  <Val>0</Val>
</DAI>
<DAI name="tOff">
  <Val>0</Val>
</DAI>
<DAI name="tOffT">
  <Val>0</Val>
</DAI>
<DAI name="minOnPer">
  <Val>30</Val>
</DAI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche17">
<DAI name="enable">
<Val>0</Val>
</DAI>
<DAI name="day">
<Val>0</Val>
</DAI>
<DAI name="tOn">
<Val>0</Val>
</DAI>
<DAI name="tOnT">
<Val>0</Val>
</DAI>
<DAI name="tOff">
<Val>0</Val>
</DAI>
<DAI name="toffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche18">
<DAI name="enable">
<Val>0</Val>
</DAI>
<DAI name="day">
<Val>0</Val>
</DAI>
<DAI name="tOn">
<Val>0</Val>
</DAI>
<DAI name="tOnT">
<Val>0</Val>
</DAI>
<DAI name="tOff">
<Val>0</Val>
</DAI>
<DAI name="toffT">
<Val>0</Val>
</DAI>
</SDI>
<DAI name="minOnPer">  
<Val>30</Val>
</DAI>
<DAI name="minOffPer">  
<Val>30</Val>
</DAI>
<DAI name="srBefWd">  
<Val>30</Val>
</DAI>
<DAI name="srAftWd">  
<Val>30</Val>
</DAI>
<DAI name="igBefWd">  
<Val>0</Val>
</DAI>
<DAI name="igAftWd">  
<Val>0</Val>
</DAI>
<DAI name="Descr">  
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche19">
<DAI name="enable">  
<Val>0</Val>
</DAI>
<DAI name="day">  
<Val>0</Val>
</DAI>
<DAI name="tOn">  
<Val>0</Val>
</DAI>
<DAI name="tOnT">  
<Val>0</Val>
</DAI>
<DAI name="tOff">  
<Val>0</Val>
</DAI>
<DAI name="tOffT">  
<Val>0</Val>
</DAI>
<DAI name="minOnPer">  
<Val>30</Val>
</DAI>
<DAI name="minOffPer">  
<Val>30</Val>
</DAI>
<DAI name="srBefWd">  
<Val>30</Val>
</DAI>
<DAI name="srAftWd">  
<Val>30</Val>
</DAI>
<DAI name="igBefWd">  
<Val>0</Val>
</DAI>
<DAI name="igAftWd">  
<Val>0</Val>
</DAI>
<DAI name="Descr">  
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
</SDI>
<SDI name="sche20">
<DAI name="enable">  
<Val>0</Val>
</DAI>
<DAI name="day">  
<Val>0</Val>
</DAI>
<DAI name="tOn">  
<Val>0</Val>
</DAI>
<SDI>
<SDI name="sche22">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
</SDI>
<SDI name="sche23">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
</SDI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>

<DAI name="srBefWd">
  <Val>30</Val>
</DAI>

<DAI name="srAftWd">
  <Val>30</Val>
</DAI>

<DAI name="igBefWd">
  <Val>0</Val>
</DAI>

<DAI name="igAftWd">
  <Val>0</Val>
</DAI>

<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>

<SDI name="sche26">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>

<SDI name="sche27">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
</SDI>
<DAI name="tOnT">
  <Val>0</Val>
</DAI>
<DAI name="tOff">
  <Val>0</Val>
</DAI>
<DAI name="tOffT">
  <Val>0</Val>
</DAI>
<DAI name="minOnPer">
  <Val>30</Val>
</DAI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche28">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche29">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>

<SDI name="sche30">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<DAI name="enable" />
<Val>0</Val>
</DAI>
<DAI name="day" />
<Val>0</Val>
</DAI>
<DAI name="tOn" />
<Val>0</Val>
</DAI>
<DAI name="tOnT" />
<Val>0</Val>
</DAI>
<DAI name="tOff" />
<Val>0</Val>
</DAI>
<DAI name="tOffT" />
<Val>0</Val>
</DAI>
<DAI name="minOnPer" />
<Val>30</Val>
</DAI>
<DAI name="minOffPer" />
<Val>30</Val>
</DAI>
<DAI name="srBefWd" />
<Val>30</Val>
</DAI>
<DAI name="srAftWd" />
<Val>30</Val>
</DAI>
<DAI name="igBefWd" />
<Val>0</Val>
</DAI>
<DAI name="igAftWd" />
<Val>0</Val>
</DAI>
<DAI name="Descr" />
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche37" >
<DAI name="enable" />
<Val>0</Val>
</DAI>
<DAI name="day" />
<Val>0</Val>
</DAI>
<DAI name="tOn" />
<Val>0</Val>
</DAI>
<DAI name="tOnT" />
<Val>0</Val>
</DAI>
<DAI name="tOff" />
<Val>0</Val>
</DAI>
<DAI name="tOffT" />
<Val>0</Val>
</DAI>
<DAI name="minOnPer" />
<Val>30</Val>
</DAI>
<DAI name="minOffPer" />
<Val>30</Val>
</DAI>
<DAI name="srBefWd" />
<Val>30</Val>
</DAI>
<DAI name="srAftWd" />
<Val>30</Val>
</DAI>
<DAI name="igBefWd" >
<Val>323</Val>
</DAI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche40">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche41">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
</SDI>
</DAI>

<DAI name="tOff">
  <Val>0</Val>
</DAI>

<DAI name="tOffT">
  <Val>0</Val>
</DAI>

<DAI name="minOnPer">
  <Val>30</Val>
</DAI>

<DAI name="minOffPer">
  <Val>30</Val>
</DAI>

<DAI name="srBefWd">
  <Val>30</Val>
</DAI>

<DAI name="srAftWd">
  <Val>30</Val>
</DAI>

<DAI name="igBefWd">
  <Val>0</Val>
</DAI>

<DAI name="igAftWd">
  <Val>0</Val>
</DAI>

<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>

</SDI>

<SDI name="sche42">
  
  </SDI>

<SDI name="sche43">
  
  </SDI>

SWDevice-010805
<Val>0</Val>
</DAI>
<DAI name="day">
<Val>0</Val>
</DAI>
<DAI name="tOn">
<Val>0</Val>
</DAI>
<DAI name="tOnT">
<Val>0</Val>
</DAI>
<DAI name="tOff">
<Val>0</Val>
</DAI>
<DAI name="tOffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche44">
<DAI name="enable">
<Val>0</Val>
</DAI>
<DAI name="day">
<Val>0</Val>
</DAI>
<DAI name="tOn">
<Val>0</Val>
</DAI>
<DAI name="tOnT">
<Val>0</Val>
</DAI>
<DAI name="tOff">
<Val>0</Val>
</DAI>
<DAI name="tOffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche47">
<DAI name="enable">
<Val>0</Val>
</DAI>
<DAI name="day">
<Val>0</Val>
</DAI>
<DAI name="tOn">
<Val>0</Val>
</DAI>
<DAI name="tOnT">
<Val>0</Val>
</DAI>
<DAI name="tOff">
<Val>0</Val>
</DAI>
<DAI name="tOffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche48">
<DAI name="enable">
<Val>0</Val>
</DAI>
<DAI name="day">
<Val>0</Val>
</DAI>
<DAI name="tOn">
<Val>0</Val>
</DAI>
<DAI name="tOnT">
<Val>0</Val>
</DAI>
<DAI name="tOff">
<Val>0</Val>
</DAI>
<DAI name="tOffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<DAI name="day">  
<Val>0</Val>  
</DAI>  
<DAI name="tOn">  
<Val>0</Val>  
</DAI>  
<DAI name="tOnT">  
<Val>0</Val>  
</DAI>  
<DAI name="tOff">  
<Val>0</Val>  
</DAI>  
<DAI name="tOffT">  
<Val>0</Val>  
</DAI>  
<DAI name="minOnPer">  
<Val>30</Val>  
</DAI>  
<DAI name="minOffPer">  
<Val>30</Val>  
</DAI>  
<DAI name="srBefWd">  
<Val>30</Val>  
</DAI>  
<DAI name="srAftWd">  
<Val>30</Val>  
</DAI>  
<DAI name="igBefWd">  
<Val>0</Val>  
</DAI>  
<DAI name="igAftWd">  
<Val>0</Val>  
</DAI>  
<DAI name="Descr">  
<Val>ON/OFF schedule</Val>  
</DAI>  
</SDI>  
<SDI name="sche51">  
<DAI name="enable">  
<Val>0</Val>  
</DAI>  
<DAI name="day">  
<Val>0</Val>  
</DAI>  
<DAI name="tOn">  
<Val>0</Val>  
</DAI>  
<DAI name="tOnT">  
<Val>0</Val>  
</DAI>  
<DAI name="tOff">  
<Val>0</Val>  
</DAI>  
<DAI name="tOffT">  
<Val>0</Val>  
</DAI>  
<DAI name="minOnPer">  
<Val>30</Val>  
</DAI>  
<DAI name="minOffPer">  
<Val>30</Val>  
</DAI>  
<DAI name="srBefWd">  
<Val>30</Val>  
</DAI>  
<DAI name="srAftWd">  
<Val>30</Val>  
</DAI>  
<DAI name="igBefWd">  
<Val>0</Val>  
</DAI>  
<DAI name="igAftWd">  
<Val>0</Val>  
</DAI>  
<DAI name="Descr">  
<Val>ON/OFF schedule</Val>  
</DAI>  
</SDI>
<DAI name="igAftWd">
    <Val>0</Val>
</DAI>
<DAI name="Descr">
    <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche52">
    <DAI name="enable">
        <Val>0</Val>
    </DAI>
    <DAI name="day">
        <Val>0</Val>
    </DAI>
    <DAI name="tOn">
        <Val>0</Val>
    </DAI>
    <DAI name="tOnT">
        <Val>0</Val>
    </DAI>
    <DAI name="tOff">
        <Val>0</Val>
    </DAI>
    <DAI name="tOffT">
        <Val>0</Val>
    </DAI>
    <DAI name="minOnPer">
        <Val>30</Val>
    </DAI>
    <DAI name="minOffPer">
        <Val>30</Val>
    </DAI>
    <DAI name="srBefWd">
        <Val>30</Val>
    </DAI>
    <DAI name="srAftWd">
        <Val>30</Val>
    </DAI>
    <DAI name="igBefWd">
        <Val>0</Val>
    </DAI>
    <DAI name="igAftWd">
        <Val>0</Val>
    </DAI>
    <DAI name="Descr">
        <Val>ON/OFF schedule</Val>
    </DAI>
</SDI>
</SDI>
<SDI name="sche53">
    <DAI name="enable">
        <Val>0</Val>
    </DAI>
    <DAI name="day">
        <Val>0</Val>
    </DAI>
    <DAI name="tOn">
        <Val>0</Val>
    </DAI>
    <DAI name="tOnT">
        <Val>0</Val>
    </DAI>
    <DAI name="tOff">
        <Val>0</Val>
    </DAI>
    <DAI name="tOffT">
        <Val>0</Val>
    </DAI>
    <DAI name="minOnPer">
        <Val>30</Val>
    </DAI>
    <DAI name="minOffPer">
        <Val>30</Val>
    </DAI>
    <DAI name="Descr">
        <Val>ON/OFF schedule</Val>
    </DAI>
</SDI>
</SDI>
<DAI name="tOffT">
  <Val>0</Val>
</DAI>
<DAI name="minOnPer">
  <Val>30</Val>
</DAI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche56">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche57">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
</SDI>
<DAI name="day">
  <Val>0</Val>
</DAI>
<DAI name="tOn">
  <Val>0</Val>
</DAI>
<DAI name="tOnT">
  <Val>0</Val>
</DAI>
<DAI name="tOff">
  <Val>0</Val>
</DAI>
<DAI name="tOffT">
  <Val>0</Val>
</DAI>
<DAI name="minOnPer">
  <Val>30</Val>
</DAI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche58">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche59">
<DAI name="enable">
<Val>0</Val>
</DAI>
</SDI>
<SDI name="sche60">
<DAI name="enable">
<Val>0</Val>
</DAI>
</SDI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche61">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="toffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche62">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="toff">
    <Val>0</Val>
  </DAI>
  <DAI name="toffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
ON/OFF schedule

SDI name="sche63"
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>

SDI name="sche64"
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<DAI name="tOn">0</DAI>
<DAI name="tOff">0</DAI>
<DAI name="tOnT">0</DAI>
<DAI name="tOffT">0</DAI>
<DAI name="minOnPer">30</DAI>
<DAI name="minOffPer">30</DAI>
<DAI name="srBefWd">30</DAI>
<DAI name="srAftWd">30</DAI>
<DAI name="igBefWd">0</DAI>
<DAI name="igAftWd">0</DAI>
<DAI name="Descr">ON/OFF schedule</DAI>
<SDI>
</SDI>
</DOI>
</LN>
<LN lnType="SwDeviceGenericI0.XSWC1" lnClass="XSWC" instn="4">
<DOI name="SwType">
<DAI name="ctlModel">status-only</DAI>
</DOI>
<DOI name="Pos">
<DAI name="ctlModel">status-only</DAI>
</DOI>
<DOI name="CfSt">
<DAI name="enbOper">1</DAI>
</DOI>
</DOI>
<DOI name="Sche">
<SDI name="sche1">
<DAI name="enable">0</DAI>
</SDI>
</DOI>
<DOI name="day">
<DAI name="tOn">0</DAI>
</DOI>
<DOI name="tOnT">
<DAI name="tOff">0</DAI>
</DOI>
<DOI name="tOffT">
<DAI name="tOff">0</DAI>
</DOI>
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
    <Val>30</Val>
</DAI>
<DAI name="minOffPer">
    <Val>30</Val>
</DAI>
<DAI name="srBefWd">
    <Val>30</Val>
</DAI>
<DAI name="srAftWd">
    <Val>30</Val>
</DAI>
<DAI name="igBefWd">
    <Val>0</Val>
</DAI>
<DAI name="igAftWd">
    <Val>0</Val>
</DAI>
<DAI name="Descr">
    <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche2">
    <DAI name="enable">
        <Val>0</Val>
    </DAI>
    <DAI name="day">
        <Val>0</Val>
    </DAI>
    <DAI name="tOn">
        <Val>0</Val>
    </DAI>
    <DAI name="tOnT">
        <Val>0</Val>
    </DAI>
    <DAI name="tOff">
        <Val>0</Val>
    </DAI>
    <DAI name="toffT">
        <Val>0</Val>
    </DAI>
    <DAI name="minOnPer">
        <Val>30</Val>
    </DAI>
    <DAI name="minOffPer">
        <Val>30</Val>
    </DAI>
    <DAI name="srBefWd">
        <Val>30</Val>
    </DAI>
    <DAI name="srAftWd">
        <Val>30</Val>
    </DAI>
    <DAI name="igBefWd">
        <Val>0</Val>
    </DAI>
    <DAI name="igAftWd">
        <Val>0</Val>
    </DAI>
    <DAI name="Descr">
        <Val>ON/OFF schedule</Val>
    </DAI>
</SDI>
<SDI name="sche3">
    <DAI name="enable">
        <Val>0</Val>
    </DAI>
    <DAI name="day">
        <Val>0</Val>
    </DAI>
</SDI>
<DAI name="tOn">  
<Val>0</Val>
</DAI>
<DAI name="tOnT">  
<Val>0</Val>
</DAI>
<DAI name="tOff">  
<Val>0</Val>
</DAI>
<DAI name="tOffT">  
<Val>0</Val>
</DAI>
<DAI name="minOnPer">  
<Val>30</Val>
</DAI>
<DAI name="minOffPer">  
<Val>30</Val>
</DAI>
<DAI name="srBefWd">  
<Val>30</Val>
</DAI>
<DAI name="srAftWd">  
<Val>30</Val>
</DAI>
<DAI name="igBefWd">  
<Val>0</Val>
</DAI>
<DAI name="igAftWd">  
<Val>0</Val>
</DAI>
<DAI name="Descr">  
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche4">  
<DAI name="enable">  
<Val>0</Val>
</DAI>
<DAI name="day">  
<Val>0</Val>
</DAI>
<DAI name="tOn">  
<Val>0</Val>
</DAI>
<DAI name="tOnT">  
<Val>0</Val>
</DAI>
<DAI name="tOff">  
<Val>0</Val>
</DAI>
<DAI name="tOffT">  
<Val>0</Val>
</DAI>
<DAI name="minOnPer">  
<Val>30</Val>
</DAI>
<DAI name="minOffPer">  
<Val>30</Val>
</DAI>
<DAI name="srBefWd">  
<Val>30</Val>
</DAI>
<DAI name="srAftWd">  
<Val>30</Val>
</DAI>
<DAI name="igBefWd">  
<Val>0</Val>
</DAI>
<DAI name="igAftWd">  
<Val>0</Val>
</DAI>
<DAI name="Descr">  
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche5">
<DAI name="enable">
<Val>0</Val>
</DAI>
<DAI name="day">
<Val>0</Val>
</DAI>
<DAI name="tOn">
<Val>0</Val>
</DAI>
<DAI name="tOnT">
<Val>0</Val>
</DAI>
<DAI name="tOff">
<Val>0</Val>
</DAI>
<DAI name="tOffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche6">
<DAI name="enable">
<Val>0</Val>
</DAI>
<DAI name="day">
<Val>0</Val>
</DAI>
<DAI name="tOn">
<Val>0</Val>
</DAI>
<DAI name="tOnT">
<Val>0</Val>
</DAI>
<DAI name="tOff">
<Val>0</Val>
</DAI>
<DAI name="tOffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche7">
<DAI name="enable">
<Val>0</Val>
</DAI>
<DAI name="day">
<Val>0</Val>
</DAI>
<DAI name="tOn">
<Val>0</Val>
</DAI>
<DAI name="tOnT">
<Val>0</Val>
</DAI>
<DAI name="tOff">
<Val>0</Val>
</DAI>
<DAI name="tOffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche8">
<DAI name="enable">
<Val>0</Val>
</DAI>
<DAI name="day">
<Val>0</Val>
</DAI>
<DAI name="tOn">
<Val>0</Val>
</DAI>
<DAI name="tOnT">
<Val>0</Val>
</DAI>
<DAI name="tOff">
<Val>0</Val>
</DAI>
<DAI name="tOffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<DAI name="minOnPer">
   <Val>30</Val>
</DAI>

<DAI name="minOffPer">
   <Val>30</Val>
</DAI>

<DAI name="srBefWd">
   <Val>30</Val>
</DAI>

<DAI name="srAftWd">
   <Val>30</Val>
</DAI>

<DAI name="igBefWd">
   <Val>0</Val>
</DAI>

<DAI name="igAftWd">
   <Val>0</Val>
</DAI>

<DAI name="Descr">
   <Val>ON/OFF schedule</Val>
</DAI>

</SDI>

<SDI name="sche9">
   <DAI name="enable">
      <Val>0</Val>
   </DAI>
   <DAI name="day">
      <Val>0</Val>
   </DAI>
   <DAI name="tOn">
      <Val>0</Val>
   </DAI>
   <DAI name="tOnT">
      <Val>0</Val>
   </DAI>
   <DAI name="tOff">
      <Val>0</Val>
   </DAI>
   <DAI name="tOffT">
      <Val>0</Val>
   </DAI>
   <DAI name="minOnPer">
      <Val>30</Val>
   </DAI>
   <DAI name="minOffPer">
      <Val>30</Val>
   </DAI>
   <DAI name="srBefWd">
      <Val>30</Val>
   </DAI>
   <DAI name="srAftWd">
      <Val>30</Val>
   </DAI>
   <DAI name="igBefWd">
      <Val>0</Val>
   </DAI>
   <DAI name="igAftWd">
      <Val>0</Val>
   </DAI>
   <DAI name="Descr">
      <Val>ON/OFF schedule</Val>
   </DAI>
</SDI>

<SDI name="sche10">
   <DAI name="enable">
      <Val>0</Val>
   </DAI>
   <DAI name="day">
      <Val>0</Val>
   </DAI>
   <DAI name="tOn">
      <Val>0</Val>
   </DAI>
   <DAI name="tOnT">
      <Val>0</Val>
   </DAI>
   <DAI name="tOff">
      <Val>0</Val>
   </DAI>
   <DAI name="tOffT">
      <Val>0</Val>
   </DAI>
   <DAI name="minOnPer">
      <Val>30</Val>
   </DAI>
   <DAI name="minOffPer">
      <Val>30</Val>
   </DAI>
   <DAI name="srBefWd">
      <Val>30</Val>
   </DAI>
   <DAI name="srAftWd">
      <Val>30</Val>
   </DAI>
   <DAI name="igBefWd">
      <Val>0</Val>
   </DAI>
   <DAI name="igAftWd">
      <Val>0</Val>
   </DAI>
   <DAI name="Descr">
      <Val>ON/OFF schedule</Val>
   </DAI>
</SDI>
<Val>0</Val>
</DAI>
<DAI name="tOnT">
<Val>0</Val>
</DAI>
<DAI name="tOff">
<Val>0</Val>
</DAI>
<DAI name="tOffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche11">
<DAI name="enable">
<Val>0</Val>
</DAI>
<DAI name="day">
<Val>0</Val>
</DAI>
<DAI name="tOn">
<Val>0</Val>
</DAI>
<DAI name="tOnT">
<Val>0</Val>
</DAI>
<DAI name="tOff">
<Val>0</Val>
</DAI>
<DAI name="tOffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<DAI/>
</SDI>
<SDI name="sche12">
<DAI name="enable">
<Val>0</Val>
</DAI>
<DAI name="day">
<Val>0</Val>
</DAI>
<DAI name="tOn">
<Val>0</Val>
</DAI>
<DAI name="tOnT">
<Val>0</Val>
</DAI>
<DAI name="tOff">
<Val>0</Val>
</DAI>
<DAI name="tOffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche13">
<DAI name="enable">
<Val>0</Val>
</DAI>
<DAI name="day">
<Val>0</Val>
</DAI>
<DAI name="tOn">
<Val>0</Val>
</DAI>
<DAI name="tOnT">
<Val>0</Val>
</DAI>
<DAI name="tOff">
<Val>0</Val>
</DAI>
<DAI name="tOffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
</SDI>
<DAI name="minOnPer">
  <Val>30</Val>
</DAI>

<DAI name="minOffPer">
  <Val>30</Val>
</DAI>

<DAI name="srBefWd">
  <Val>30</Val>
</DAI>

<DAI name="srAftWd">
  <Val>30</Val>
</DAI>

<DAI name="igBefWd">
  <Val>0</Val>
</DAI>

<DAI name="igAftWd">
  <Val>0</Val>
</DAI>

<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>

</SDI>

<SDI name="sche16">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>

  <DAI name="day">
    <Val>0</Val>
  </DAI>

  <DAI name="tOn">
    <Val>0</Val>
  </DAI>

  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>

  <DAI name="tOff">
    <Val>0</Val>
  </DAI>

  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>

  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>

  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>

  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>

  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>

  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>

  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>

  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>

</SDI>

</SDI>

<SDI name="sche17">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>

  <DAI name="day">
    <Val>0</Val>
  </DAI>

  <DAI name="tOn">
    <Val>0</Val>
  </DAI>

</SDI>
<DAI>
  <DAI name="tOnT">  
    <Val>0</Val>  
  </DAI>
  <DAI name="tOff">  
    <Val>0</Val>  
  </DAI>
  <DAI name="tOffT">  
    <Val>0</Val>  
  </DAI>
  <DAI name="minOnPer">  
    <Val>30</Val>  
  </DAI>
  <DAI name="minOffPer">  
    <Val>30</Val>  
  </DAI>
  <DAI name="srBefWd">  
    <Val>30</Val>  
  </DAI>
  <DAI name="srAftWd">  
    <Val>30</Val>  
  </DAI>
  <DAI name="igBefWd">  
    <Val>0</Val>  
  </DAI>
  <DAI name="igAftWd">  
    <Val>0</Val>  
  </DAI>
  <DAI name="Descr">  
    <Val>ON/OFF schedule</Val>  
  </DAI>
</DAI>
</SDI>

<SDI name="sche18">  
  <DAI name="enable">  
    <Val>0</Val>  
  </DAI>
  <DAI name="day">  
    <Val>0</Val>  
  </DAI>
  <DAI name="tOn">  
    <Val>0</Val>  
  </DAI>
  <DAI name="tOnT">  
    <Val>0</Val>  
  </DAI>
  <DAI name="tOff">  
    <Val>0</Val>  
  </DAI>
  <DAI name="tOffT">  
    <Val>0</Val>  
  </DAI>
  <DAI name="minOnPer">  
    <Val>30</Val>  
  </DAI>
  <DAI name="minOffPer">  
    <Val>30</Val>  
  </DAI>
  <DAI name="srBefWd">  
    <Val>30</Val>  
  </DAI>
  <DAI name="srAftWd">  
    <Val>30</Val>  
  </DAI>
  <DAI name="igBefWd">  
    <Val>0</Val>  
  </DAI>
  <DAI name="igAftWd">  
    <Val>0</Val>  
  </DAI>
  <DAI name="Descr">  
    <Val>ON/OFF schedule</Val>  
  </DAI>
</SDI>
<DAI name="igBefWd">  
<Val>0</Val>  
</DAI>
  
<DAI name="igAftWd">  
<Val>0</Val>  
</DAI>
  
<DAI name="Descr">  
<Val>ON/OFF schedule</Val>  
</DAI>
</SDI>

<SDI name="sche21">
  
<DAI name="enable">  
<Val>0</Val>  
</DAI>
  
<DAI name="day">  
<Val>0</Val>  
</DAI>
  
<DAI name="tOn">  
<Val>0</Val>  
</DAI>
  
<DAI name="tOnT">  
<Val>0</Val>  
</DAI>
  
<DAI name="tOff">  
<Val>0</Val>  
</DAI>
  
<DAI name="tOffT">  
<Val>0</Val>  
</DAI>
  
<DAI name="minOnPer">  
<Val>30</Val>  
</DAI>
  
<DAI name="minOffPer">  
<Val>30</Val>  
</DAI>
  
<DAI name="srBefWd">  
<Val>30</Val>  
</DAI>
  
<DAI name="srAftWd">  
<Val>30</Val>  
</DAI>
  
<DAI name="igBefWd">  
<Val>0</Val>  
</DAI>
  
<DAI name="igAftWd">  
<Val>0</Val>  
</DAI>
  
<DAI name="Descr">  
<Val>ON/OFF schedule</Val>  
</DAI>
</SDI>

<SDI name="sche22">
  
<DAI name="enable">  
<Val>0</Val>  
</DAI>
  
<DAI name="day">  
<Val>0</Val>  
</DAI>
  
<DAI name="tOn">  
<Val>0</Val>  
</DAI>
  
<DAI name="tOnT">  
<Val>0</Val>  
</DAI>
  
<DAI name="tOff">  
<Val>0</Val>  
</DAI>
  
<DAI name="tOffT">  
<Val>0</Val>  
</DAI>
  
<DAI name="minOnPer">  
<Val>30</Val>  
</DAI>
</SDI>
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche23">
<DAI name="enable">
<Val>0</Val>
</DAI>
<DAI name="day">
<Val>0</Val>
</DAI>
<DAI name="tOn">
<Val>0</Val>
</DAI>
<DAI name="tOnT">
<Val>0</Val>
</DAI>
<DAI name="tOff">
<Val>0</Val>
</DAI>
<DAI name="tOffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche24">
<DAI name="enable">
<Val>0</Val>
</DAI>
<DAI name="day">
<Val>0</Val>
</DAI>
<DAI name="tOn">
<Val>0</Val>
</DAI>
</DAI>
</SDI>
</SWDevice-010805>
<SDI name="sche26">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>

<SDI name="sche27">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
</SDI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche28">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche29">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
</SDI>
ON/OFF schedule

Grid eXchange Fabric Documentation
<DAI name="enable">
    <Val>0</Val>
</DAI>
<DAI name="day">
    <Val>0</Val>
</DAI>
<DAI name="tOn">
    <Val>0</Val>
</DAI>
<DAI name="tOnT">
    <Val>0</Val>
</DAI>
<DAI name="tOff">
    <Val>0</Val>
</DAI>
<DAI name="tOffT">
    <Val>0</Val>
</DAI>
<DAI name="minOnPer">
    <Val>30</Val>
</DAI>
<DAI name="minOffPer">
    <Val>30</Val>
</DAI>
<DAI name="srBefWd">
    <Val>30</Val>
</DAI>
<DAI name="srAftWd">
    <Val>30</Val>
</DAI>
<DAI name="igBefWd">
    <Val>0</Val>
</DAI>
<DAI name="igAftWd">
    <Val>0</Val>
</DAI>
<DAI name="Descr">
    <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche34">
    <DAI name="enable">
        <Val>0</Val>
    </DAI>
    <DAI name="day">
        <Val>0</Val>
    </DAI>
    <DAI name="tOn">
        <Val>0</Val>
    </DAI>
    <DAI name="tOnT">
        <Val>0</Val>
    </DAI>
    <DAI name="tOff">
        <Val>0</Val>
    </DAI>
    <DAI name="tOffT">
        <Val>0</Val>
    </DAI>
    <DAI name="minOnPer">
        <Val>30</Val>
    </DAI>
    <DAI name="minOffPer">
        <Val>30</Val>
    </DAI>
    <DAI name="srBefWd">
        <Val>30</Val>
    </DAI>
    <DAI name="srAftWd">
        <Val>30</Val>
    </DAI>
    <DAI name="igBefWd">
        <Val>0</Val>
    </DAI>
    <DAI name="igAftWd">
        <Val>0</Val>
    </DAI>
    <DAI name="Descr">
        <Val>ON/OFF schedule</Val>
    </DAI>
</SDI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
<SDI name="sche35">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche36">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
</SDI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche37">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche38">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
<DAI name="tOff">
  <Val>0</Val>
</DAI>
<DAI name="tOffT">
  <Val>0</Val>
</DAI>
<DAI name="minOnPer">
  <Val>30</Val>
</DAI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche39">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="ton">
    <Val>0</Val>
  </DAI>
  <DAI name="tonT">
    <Val>0</Val>
  </DAI>
  <DAI name="toff">
    <Val>0</Val>
  </DAI>
  <DAI name="toffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche40">
  <DAI name="enable">
    <Val>false</Val>
  </DAI>
</SDI>
<Val>0</Val>
</DAI>
<DAI name="day">
<Val>0</Val>
</DAI>
<DAI name="tOn">
<Val>0</Val>
</DAI>
<DAI name="tOnT">
<Val>0</Val>
</DAI>
<DAI name="tOff">
<Val>0</Val>
</DAI>
<DAI name="tOffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche41">
<DAI name="enable">
<Val>0</Val>
</DAI>
<DAI name="day">
<Val>0</Val>
</DAI>
<DAI name="tOn">
<Val>0</Val>
</DAI>
<DAI name="tOnT">
<Val>0</Val>
</DAI>
<DAI name="tOff">
<Val>0</Val>
</DAI>
<DAI name="tOffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<DAI name="igAftWd"> <Val>0</Val> </DAI>
<DAI name="Descr"> <Val>ON/OFF schedule</Val> </DAI>
</SDI>
<SDI name="sche42">
<DAI name="enable"> <Val>0</Val> </DAI>
<DAI name="day"> <Val>0</Val> </DAI>
<DAI name="tOn"> <Val>0</Val> </DAI>
<DAI name="tOnT"> <Val>0</Val> </DAI>
<DAI name="tOff"> <Val>0</Val> </DAI>
<DAI name="tOffT"> <Val>0</Val> </DAI>
<DAI name="minOnPer"> <Val>30</Val> </DAI>
<DAI name="minOffPer"> <Val>30</Val> </DAI>
<DAI name="srBefWd"> <Val>30</Val> </DAI>
<DAI name="srAftWd"> <Val>30</Val> </DAI>
<DAI name="igBefWd"> <Val>0</Val> </DAI>
<DAI name="igAftWd"> <Val>0</Val> </DAI>
<DAI name="Descr"> <Val>ON/OFF schedule</Val> </DAI>
</SDI>
<SDI name="sche43">
<DAI name="enable"> <Val>0</Val> </DAI>
<DAI name="day"> <Val>0</Val> </DAI>
<DAI name="tOn"> <Val>0</Val> </DAI>
<DAI name="tOnT"> <Val>0</Val> </DAI>
<DAI name="tOff"> <Val>0</Val> </DAI>
<DAI name="tOffT"> <Val>0</Val> </DAI>
<DAI name="minOnPer"> <Val>30</Val> </DAI>
<DAI name="minOffPer"> <Val>30</Val> </DAI>
<DAI name="srBefWd"> <Val>30</Val> </DAI>
<DAI name="srAftWd"> <Val>30</Val> </DAI>
<DAI name="igBefWd"> <Val>0</Val> </DAI>
<DAI name="igAftWd"> <Val>0</Val> </DAI>
<DAI name="Descr"> <Val>ON/OFF schedule</Val> </DAI>
</SDI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche44">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche45">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
</DAI>
<DAI name="tOff">
  <Val>0</Val>
</DAI>
<DAI name="tOffT">
  <Val>0</Val>
</DAI>
<DAI name="minOnPer">
  <Val>30</Val>
</DAI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche46">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche47">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
</SDI>
<DAI name="day">  
<Val>0</Val>  
</DAI>
<DAI name="tOn">  
<Val>0</Val>  
</DAI>
<DAI name="tOnT">  
<Val>0</Val>  
</DAI>
<DAI name="tOff">  
<Val>0</Val>  
</DAI>
<DAI name="tOffT">  
<Val>0</Val>  
</DAI>
<DAI name="minOnPer">  
<Val>30</Val>  
</DAI>
<DAI name="minOffPer">  
<Val>30</Val>  
</DAI>
<DAI name="srBefWd">  
<Val>30</Val>  
</DAI>
<DAI name="srAftWd">  
<Val>30</Val>  
</DAI>
<DAI name="igBefWd">  
<Val>0</Val>  
</DAI>
<DAI name="igAftWd">  
<Val>0</Val>  
</DAI>
<DAI name="Descr">  
<Val>ON/OFF schedule</Val>  
</DAI>
</SDI>
<SDI name="sche48">  
<DAI name="enable">  
<Val>0</Val>  
</DAI>
<DAI name="day">  
<Val>0</Val>  
</DAI>
<DAI name="tOn">  
<Val>0</Val>  
</DAI>
<DAI name="tOnT">  
<Val>0</Val>  
</DAI>
<DAI name="tOff">  
<Val>0</Val>  
</DAI>
<DAI name="tOffT">  
<Val>0</Val>  
</DAI>
<DAI name="minOnPer">  
<Val>30</Val>  
</DAI>
<DAI name="minOffPer">  
<Val>30</Val>  
</DAI>
<DAI name="srBefWd">  
<Val>30</Val>  
</DAI>
<DAI name="srAftWd">  
<Val>30</Val>  
</DAI>
<DAI name="igBefWd">  
<Val>0</Val>  
</DAI>
<DAI name="igAftWd">  
<Val>0</Val>  
</DAI>
<DAI name="Descr">  
<Val>ON/OFF schedule</Val>  
</DAI>
</SDI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche49">
<DAI name="enable">
<Val>0</Val>
</DAI>
<DAI name="day">
<Val>0</Val>
</DAI>
<DAI name="tOn">
<Val>0</Val>
</DAI>
<DAI name="tOnT">
<Val>0</Val>
</DAI>
<DAI name="tOff">
<Val>0</Val>
</DAI>
<DAI name="tOffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
<DAI name="srBefWd">
<Val>30</Val>
</DAI>
<DAI name="srAftWd">
<Val>30</Val>
</DAI>
<DAI name="igBefWd">
<Val>0</Val>
</DAI>
<DAI name="igAftWd">
<Val>0</Val>
</DAI>
<DAI name="Descr">
<Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche50">
<DAI name="enable">
<Val>0</Val>
</DAI>
<DAI name="day">
<Val>0</Val>
</DAI>
<DAI name="tOn">
<Val>0</Val>
</DAI>
<DAI name="tOnT">
<Val>0</Val>
</DAI>
<DAI name="tOff">
<Val>0</Val>
</DAI>
<DAI name="tOffT">
<Val>0</Val>
</DAI>
<DAI name="minOnPer">
<Val>30</Val>
</DAI>
<DAI name="minOffPer">
<Val>30</Val>
</DAI>
</SDI>
ON/OFF schedule

<DAI name="srBefWd"> <Val>30</Val> </DAI>
<DAI name="srAftWd"> <Val>30</Val> </DAI>
<DAI name="igBefWd"> <Val>0</Val> </DAI>
<DAI name="igAftWd"> <Val>0</Val> </DAI>
<DAI name="Descr"> <Val>ON/OFF schedule</Val> </DAI>

<DAI name="sche51"> <DAI name="enable"> <Val>0</Val> </DAI> <DAI name="day"> <Val>0</Val> </DAI> <DAI name="tOn"> <Val>0</Val> </DAI> <DAI name="tOnT"> <Val>0</Val> </DAI> <DAI name="tOff"> <Val>0</Val> </DAI> <DAI name="tOffT"> <Val>0</Val> </DAI> <DAI name="minOnPer"> <Val>30</Val> </DAI> <DAI name="minOffPer"> <Val>30</Val> </DAI> <DAI name="srBefWd"> <Val>30</Val> </DAI> <DAI name="srAftWd"> <Val>30</Val> </DAI> <DAI name="igBefWd"> <Val>0</Val> </DAI> <DAI name="igAftWd"> <Val>0</Val> </DAI> <DAI name="Descr"> <Val>ON/OFF schedule</Val> </DAI>

<SDI> <SDI name="sche51"> <DAI name="enable"> <Val>0</Val> </DAI> </SDI> </SDI>

<SDI name="sche52"> <DAI name="enable"> <Val>0</Val> </DAI> <DAI name="day"> <Val>0</Val> </DAI> <DAI name="tOn"> <Val>0</Val> </DAI> <DAI name="tOnT"> <Val>0</Val> </DAI> <DAI name="tOff"> <Val>0</Val> </DAI> <DAI name="tOffT"> <Val>0</Val> </DAI> <DAI name="minOnPer"> <Val>30</Val> </DAI> <DAI name="minOffPer"> <Val>30</Val> </DAI> <DAI name="srBefWd"> <Val>30</Val> </DAI> <DAI name="srAftWd"> <Val>30</Val> </DAI> <DAI name="igBefWd"> <Val>0</Val> </DAI> <DAI name="igAftWd"> <Val>0</Val> </DAI> <DAI name="Descr"> <Val>ON/OFF schedule</Val> </DAI>
</SDI>

</SDI>
<DAI name="day">
  <Val>0</Val>
</DAI>
<DAI name="tOn">
  <Val>0</Val>
</DAI>
<DAI name="tOnT">
  <Val>0</Val>
</DAI>
<DAI name="tOff">
  <Val>0</Val>
</DAI>
<DAI name="tOffT">
  <Val>0</Val>
</DAI>
<DAI name="minOnPer">
  <Val>30</Val>
</DAI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche55">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<Val>0</Val>
</DAI>
<DAI name="Descr">
 <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche56">
 <DAI name="enable">
  <Val>0</Val>
 </DAI>
 <DAI name="day">
  <Val>0</Val>
 </DAI>
 <DAI name="tOn">
  <Val>0</Val>
 </DAI>
 <DAI name="tOnT">
  <Val>0</Val>
 </DAI>
 <DAI name="tOff">
  <Val>0</Val>
 </DAI>
 <DAI name="tOffT">
  <Val>0</Val>
 </DAI>
 <DAI name="minOnPer">
  <Val>30</Val>
 </DAI>
 <DAI name="minOffPer">
  <Val>30</Val>
 </DAI>
 <DAI name="srBefWd">
  <Val>30</Val>
 </DAI>
 <DAI name="srAftWd">
  <Val>30</Val>
 </DAI>
 <DAI name="igBefWd">
  <Val>0</Val>
 </DAI>
 <DAI name="igAftWd">
  <Val>0</Val>
 </DAI>
 <DAI name="Descr">
  <Val>ON/OFF schedule</Val>
 </DAI>
</SDI>
<SDI name="sche57">
 <DAI name="enable">
  <Val>0</Val>
 </DAI>
 <DAI name="day">
  <Val>0</Val>
 </DAI>
 <DAI name="tOn">
  <Val>0</Val>
 </DAI>
 <DAI name="tOnT">
  <Val>0</Val>
 </DAI>
 <DAI name="tOff">
  <Val>0</Val>
 </DAI>
 <DAI name="tOffT">
  <Val>0</Val>
 </DAI>
 <DAI name="minOnPer">
  <Val>30</Val>
 </DAI>
 <DAI name="minOffPer">
  <Val>30</Val>
 </DAI>
</SDI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche58">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche59">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
<DAI name="tOffT">
  <Val>0</Val>
</DAI>
<DAI name="minOnPer">
  <Val>30</Val>
</DAI>
<DAI name="minOffPer">
  <Val>30</Val>
</DAI>
<DAI name="srBefWd">
  <Val>30</Val>
</DAI>
<DAI name="srAftWd">
  <Val>30</Val>
</DAI>
<DAI name="igBefWd">
  <Val>0</Val>
</DAI>
<DAI name="igAftWd">
  <Val>0</Val>
</DAI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>
</SDI>
<SDI name="sche60">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<SDI name="sche61">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
  </DAI>
  <DAI name="tOff">
  </DAI>
  <DAI name="tOffT">
  </DAI>
  <DAI name="minOnPer">
  </DAI>
  <DAI name="minOffPer">
  </DAI>
  <DAI name="srBefWd">
  </DAI>
  <DAI name="srAftWd">
  </DAI>
  <DAI name="igBefWd">
  </DAI>
  <DAI name="igAftWd">
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>
<DAI name="Descr">
  <Val>ON/OFF schedule</Val>
</DAI>

<SDI name="sche63">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
  <DAI name="srAftWd">
    <Val>30</Val>
  </DAI>
  <DAI name="igBefWd">
    <Val>0</Val>
  </DAI>
  <DAI name="igAftWd">
    <Val>0</Val>
  </DAI>
  <DAI name="Descr">
    <Val>ON/OFF schedule</Val>
  </DAI>
</SDI>

<SDI name="sche64">
  <DAI name="enable">
    <Val>0</Val>
  </DAI>
  <DAI name="day">
    <Val>0</Val>
  </DAI>
  <DAI name="tOn">
    <Val>0</Val>
  </DAI>
  <DAI name="tOnT">
    <Val>0</Val>
  </DAI>
  <DAI name="tOff">
    <Val>0</Val>
  </DAI>
  <DAI name="tOffT">
    <Val>0</Val>
  </DAI>
  <DAI name="minOnPer">
    <Val>30</Val>
  </DAI>
  <DAI name="minOffPer">
    <Val>30</Val>
  </DAI>
  <DAI name="srBefWd">
    <Val>30</Val>
  </DAI>
</SDI>
<DOType id="SWDeviceGenericIO.CSLC.FuncFwDw" cdc="UNKNOWN">
  <DA name="status" fc="ST" bType="INT8" />
  <DA name="curVer" fc="ST" bType="VisString32" />
  <DA name="url1" fc="CF" bType="VisString255" />
  <DA name="startT" fc="CF" bType="Timestamp" />
</DOType>

<DOType id="SWDeviceGenericIO.CSLC.Clock" cdc="UNKNOWN">
  <DA name="curT" fc="CF" bType="Timestamp" />
  <DA name="tz" fc="CF" bType="INT16" />
  <DA name="dstBegT" fc="CF" bType="VisString255" />
  <DA name="dvt" fc="CF" bType="INT16" />
  <DA name="enbDst" fc="CF" bType="BOOLEAN" />
  <DA name="enbNtpC" fc="CF" bType="BOOLEAN" />
  <DA name="ntpSvrA" fc="CF" bType="VisString255" />
  <DA name="syncPer" fc="CF" bType="INT16U" />
  <DA name="almNost" fc="CF" bType="INT16U" />
  <DA name="almOst" fc="CF" bType="INT16U" />
</DOType>

<DOType id="SWDeviceGenericIO.CSLC.Reg" cdc="UNKNOWN">
  <DA name="svrAddr" fc="CF" bType="VisString64" />
  <DA name="svrPort" fc="CF" bType="INT32" />
  <DA name="ntfEnb" fc="CF" bType="BOOLEAN" />
</DOType>

<DOType id="SWDeviceGenericIO.CSLC.Log" cdc="UNKNOWN">
  <DA name="enb" fc="CF" bType="BOOLEAN" />
</DOType>

<DOType id="SWDeviceGenericIO.CSLC.IPCf" cdc="UNKNOWN">
  <DA name="mac" fc="ST" bType="VisString32" />
  <DA name="enbDHCp" fc="CF" bType="BOOLEAN" />
  <DA name="ipAddr" fc="CF" bType="VisString32" />
  <DA name="netmask" fc="CF" bType="VisString32" />
  <DA name="gateway" fc="CF" bType="VisString32" />
</DOType>

<DOType id="SWDeviceGenericIO.CSLC.EvnRpn" cdc="UNKNOWN">
  <DA name="evnType" fc="ST" bType="INT8U" />
  <DA name="trgType" fc="ST" bType="INT8U" />
  <DA name="swVal1" fc="ST" bType="BOOLEAN" />
  <DA name="trgTime" fc="ST" bType="Timestamp" />
  <DA name="remark" fc="ST" bType="VisString64" />
</DOType>

<DOType id="SWDeviceGenericIO.CSLC.EvnBuf" cdc="UNKNOWN">
  <DA name="lastIdx" fc="ST" bType="INT16U" />
  <DA name="evn1" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.evn1" />
  <DA name="evn2" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.evn1" />
  <DA name="evn3" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.evn1" />
  <DA name="evn4" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.evn1" />
  <DA name="evn5" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.evn1" />
  <DA name="evn6" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.evn1" />
  <DA name="evn7" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.evn1" />
  <DA name="evn8" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.evn1" />
  <DA name="evn9" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.evn1" />
  <DA name="evn10" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.evn1" />
  <DA name="evn11" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.evn1" />
  <DA name="evn12" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.evn1" />
  <DA name="evn13" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.evn1" />
  <DA name="evn14" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.evn1" />
  <DA name="evn15" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.evn1" />
  <DA name="evn16" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.evn1" />
  <DA name="evn17" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.evn1" />
  <DA name="evn18" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.evn1" />
  <DA name="evn19" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.evn1" />
  <DA name="evn20" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.evn1" />
  <DA name="evn21" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.evn1" />
  <DA name="evn22" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.evn1" />
  <DA name="evn23" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.evn1" />
  <DA name="evn24" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.evn1" />
</DOType>
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>evn25</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn26</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn27</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn28</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn29</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn30</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn31</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn32</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn33</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn34</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn35</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn36</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn37</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn38</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn39</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn40</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn41</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn42</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn43</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn44</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn45</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn46</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn47</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn48</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn49</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn50</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn51</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn52</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn53</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn54</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn55</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn56</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn57</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn58</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn59</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn60</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn61</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn62</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn63</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn64</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn65</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn66</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn67</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn68</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn69</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn70</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn71</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn72</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn73</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn74</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn75</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn76</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn77</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn78</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn79</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn80</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn81</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn82</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn83</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn84</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn85</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn86</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn87</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn88</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn89</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn90</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn91</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn92</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn93</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn94</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn95</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
<tr>
<td>evn96</td>
<td>fc=ST bType=Struct</td>
<td>type=SWDeviceGenericIO.CSLC.EvnBuf.evn</td>
</tr>
</tbody>
</table>
<DA name="env97" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.env"
<DA name="env98" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.env"
<DA name="env99" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.env"
<DA name="env100" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.env"
<DA name="env101" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.env"
<DA name="env102" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.env"
<DA name="env103" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.env"
<DA name="env104" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.env"
<DA name="env105" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.env"
<DA name="env106" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.env"
<DA name="env107" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.env"
<DA name="env108" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.env"
<DA name="env109" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.env"
<DA name="env110" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.env"
<DA name="env111" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.env"
<DA name="env112" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.env"
<DA name="env113" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.env"
<DA name="env114" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.env"
<DA name="env115" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.env"
<DA name="env116" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.env"
<DA name="env117" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.env"
<DA name="env118" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.env"
<DA name="env119" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.env"
<DA name="env120" fc="ST" bType="Struct" type="SWDeviceGenericIO.CSLC.EvnBuf.env"
<DA name="enbEvnType" fc="CF" bType="VisString32"
</DOType>
<DOType id="SWDeviceGenericIO.CSLC.SNMPcf" cdc="UNKNOWN">
<DA name="userName" fc="CF" bType="VisString255" />
<DA name="encryptkey" fc="CF" bType="VisString255" />
</DOType>
<DOType id="SWDeviceGenericIO.XSWC1.OnItvB" cdc="UNKNOWN">
<DA name="lastIdx" fc="ST" bType="INT8u" />
<DA name="itv1" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv2" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv3" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv4" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv5" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv6" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv7" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv8" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv9" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv10" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv11" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv12" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv13" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv14" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv15" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv16" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv17" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv18" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv19" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv20" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv21" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv22" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv23" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv24" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv25" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv26" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv27" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv28" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv29" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv30" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv31" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv32" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv33" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv34" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv35" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv36" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv37" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv38" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DA name="itv39" fc="ST" bType="Struct" type="SWDeviceGenericIO.XSWC1.OnItvB.itv1"
<DOCTYPE id="SWDeviceGenericIO.LPHD.Phynam" cdc="DPL">
  <DA name="hwRev" fc="DC" bType="VisString255" />
  <DA name="serNum" fc="DC" bType="VisString255" />
  <DA name="model" fc="DC" bType="VisString255" />
  <DA name="mshn" fc="DC" bType="VisString255" />
</DOCTYPE>

<DOCTYPE id="SWDeviceGenericIO.LLN0.Mod" cdc="INC">
  <DA name="stVal" fc="ST" bType="Enum" type="Mod" />
  <DA name="q" fc="ST" bType="Quality" />
  <DA name="t" fc="ST" bType="Timestamp" />
  <DA name="Oper" fc="CO" bType="Struct" type="SWDeviceGenericIO.LLN0.Mod.Oper" />
  <DA name="ctlModel" fc="CF" bType="Enum" type="ctlModel" />
</DOCTYPE>

<DOCTYPE id="SWDeviceGenericIO.XSWC1.SwType" cdc="INC">
  <DA name="stVal" fc="ST" bType="Int8" />
  <DA name="q" fc="ST" bType="Quality" />
  <DA name="t" fc="ST" bType="Timestamp" />
  <DA name="Oper" fc="CO" bType="Struct" type="SWDeviceGenericIO.XSWC1.SwType.Oper" />
  <DA name="ctlModel" fc="CF" bType="Enum" type="ctlModel" />
</DOCTYPE>

<DOCTYPE id="SWDeviceGenericIO.LLN0.Beh" cdc="INS">
  <DA name="stVal" fc="ST" bType="Enum" type="Beh" />
  <DA name="q" fc="ST" bType="Quality" />
  <DA name="t" fc="ST" bType="Timestamp" />
</DOCTYPE>

<DOCTYPE id="SWDeviceGenericIO.LLN0.Health" cdc="INS">
  <DA name="stVal" fc="ST" bType="Enum" type="Health" />
  <DA name="q" fc="ST" bType="Quality" />
  <DA name="t" fc="ST" bType="Timestamp" />
</DOCTYPE>

<DOCTYPE id="SWDeviceGenericIO.XSWC1.OpCnt" cdc="INS">
  <DA name="stVal" fc="ST" bType="Int32" />
  <DA name="q" fc="ST" bType="Quality" />
  <DA name="t" fc="ST" bType="Timestamp" />
</DOCTYPE>

<DOCTYPE id="SWDeviceGenericIO.LLN0.NamPlt" cdc="LPL">
  <DA name="vendor" fc="DC" bType="VisString255" />
  <DA name="swRev" fc="DC" bType="VisString255" />
  <DA name="d" fc="DC" bType="VisString255" />
  <DA name="configRev" fc="DC" bType="VisString255" />
  <DA name="ldNs" fc="EX" bType="VisString255" />
</DOCTYPE>

<DOCTYPE id="SWDeviceGenericIO.CSLC.RbOper" cdc="SPC">
  <DA name="stVal" fc="ST" bType="BOOLEAN" />
  <DA name="Oper" fc="CO" bType="Struct" type="SWDeviceGenericIO.CSLC.RbOper.Oper" />
  <DA name="ctlModel" fc="CF" bType="Enum" type="ctlModel" />
</DOCTYPE>

<DOCTYPE id="SWDeviceGenericIO.CSLC.Sensor" cdc="SPC">
  <DA name="stVal" fc="ST" bType="BOOLEAN" />
  <DA name="q" fc="ST" bType="Quality" />
  <DA name="t" fc="ST" bType="Timestamp" />
</DOCTYPE>
<DA name="Oper" fc="CO" bType="Struct" type="SWDeviceGenericIO.CSLC.RbOper.Oper" />
<DA name="ctlModel" fc="CF" bType="Enum" type="ctlModel" />
</DOType>

<DOType id="SWDeviceGenericIO.LPHD.Proxy"cdc="SPS">
<DA name="stVal" fc="ST" bType="BOOLEAN" />
<DA name="q" fc="ST" bType="Quality" />
<DA name="t" fc="ST" bType="Timestamp" />
</DOType>

<DAType id="SWDeviceGenericIO.CSLC.EvnBuf.evn1">
<BA name="evnType" bType="INT8U" />
<BA name="swNum" bType="INT8U" />
<BA name="trgType" bType="INT8U" />
<BA name="swVal" bType="BOOLEAN" />
<BA name="trgTime" bType="Timestamp" />
<BA name="remark" bType="VisString64" />
</DAType>

<DAType id="SWDeviceGenericIO.XSWC1.OnItvB.itv1">
<BA name="itv" bType="INT32" />
<BA name="day" bType="Timestamp" />
</DAType>

<DAType id="SWDeviceGenericIO.XSWC1.OdItvB.itv1">
<BA name="enable" bType="BOOLEAN" />
<BA name="day" bType="INT32" />
<BA name="on" bType="INT32" />
<BA name="onT" bType="INT8" />
<BA name="off" bType="INT32" />
<BA name="offT" bType="INT8" />
<BA name="minOnPer" bType="INT16U" />
<BA name="minOffPer" bType="INT16U" />
<BA name="srBefWd" bType="INT16U" />
<BA name="igBefWd" bType="INT16U" />
<BA name="srAftWd" bType="INT16U" />
<BA name="igAftWd" bType="INT16U" />
<BA name="Descr" bType="VisString255" />
</DAType>

<DAType id="SWDeviceGenericIO.XSWC1.Oper.origin">
<BA name="orCat" bType="Enum" type="orCategory" />
<BA name="orIdent" bType="Octet64" />
</DAType>

<DAType id="SWDeviceGenericIO.LLN0.Mod.Oper.origin">
<BA name="origin" bType="Struct" type="SWDeviceGenericIO.LLN0.Mod.Oper.origin" />
<BA name="ctlNum" bType="INT8U" />
<BA name="T" bType="Timestamp" />
<BA name="Test" bType="BOOLEAN" />
<BA name="Check" bType="Check" />
</DAType>

<DAType id="SWDeviceGenericIO.LLN0.Mod.Oper">
<BA name="ctlVal1" bType="Enum" type="Mod" />
<BA name="origin" bType="Struct" type="SWDeviceGenericIO.LLN0.Mod.Oper.origin" />
<BA name="ctlNum" bType="INT8U" />
<BA name="T" bType="Timestamp" />
<BA name="Test" bType="BOOLEAN" />
<BA name="Check" bType="Check" />
</DAType>

<DAType id="SWDeviceGenericIO.XSWC1.SwType.Oper">
<BA name="ctlVal1" bType="INT8" />
<BA name="origin" bType="Struct" type="SWDeviceGenericIO.LLN0.Mod.Oper.origin" />
<BA name="ctlNum" bType="INT8U" />
<BA name="T" bType="Timestamp" />
<BA name="Test" bType="BOOLEAN" />
<BA name="Check" bType="Check" />
</DAType>

<EnumType id="ctlModel">
<!-- Source: IEC 61850-7-3:2003 -->
<EnumVal ord="0">status-only</EnumVal>
<EnumVal ord="1">direct-with-normal-security</EnumVal>
<EnumVal ord="2">sbo-with-normal-security</EnumVal>
<EnumVal ord="3">direct-with-enhanced-security</EnumVal>
<EnumVal ord="4">sbo-with-enhanced-security</EnumVal>
</EnumType>

<EnumType id="orCategory">
<!--Source: IEC 61850-7-3:2003-->  
<EnumVal ord="0">not-supported</EnumVal>  
<EnumVal ord="1">bay-control</EnumVal>  
<EnumVal ord="2">station-control</EnumVal>  
<EnumVal ord="3">remote-control</EnumVal>  
<EnumVal ord="4">automatic-bay</EnumVal>  
<EnumVal ord="5">automatic-station</EnumVal>  
<EnumVal ord="6">automatic-remote</EnumVal>  
<EnumVal ord="7">maintenance</EnumVal>  
<EnumVal ord="8">process</EnumVal>  
</EnumType>  
<EnumType id="Beh">  
<!--Source: IEC 61850-7-4:2003-->  
<EnumVal ord="1">on</EnumVal>  
<EnumVal ord="2">blocked</EnumVal>  
<EnumVal ord="3">test</EnumVal>  
<EnumVal ord="4">test/blocked</EnumVal>  
<EnumVal ord="5">off</EnumVal>  
</EnumType>  
<EnumType id="Health">  
<!--Source: IEC 61850-7-4:2003-->  
<EnumVal ord="1">Ok</EnumVal>  
<EnumVal ord="2">Warning</EnumVal>  
<EnumVal ord="3">Alarm</EnumVal>  
</EnumType>  
<EnumType id="Mod">  
<!--Source: IEC 61850-7-4:2003-->  
<EnumVal ord="1">on</EnumVal>  
<EnumVal ord="2">blocked</EnumVal>  
<EnumVal ord="3">test</EnumVal>  
<EnumVal ord="4">test/blocked</EnumVal>  
<EnumVal ord="5">off</EnumVal>  
</EnumType>  
</DataTypeTemplates>  
</SCL>
RegisterDevice

RegisterDevice messages

Description

The device registration is a 2 step process. First RegisterDeviceRequest is sent from device to platform. Second are writing GPS coordinates to the device and disabling the device registration flag.

Request that notifies the platform that a device wants to register. During the registration the device identification (serial number) and the IP address are sent to the platform.

Response writes GPS coordinates and disables registration flag.

IEC61850 Fields

<table>
<thead>
<tr>
<th>ATTRIBUTE FC</th>
<th>SUB ATTRIBUTE DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSLC.Reg</td>
<td>CF ntEnb</td>
<td>BOOLEAN</td>
</tr>
<tr>
<td>CSLC.Atnm</td>
<td>CF lon</td>
<td>FLOAT32</td>
</tr>
<tr>
<td>CSLC.Atnm</td>
<td>CF lat</td>
<td>FLOAT32</td>
</tr>
</tbody>
</table>

Plain text registration message

0000000053,84.30.69.148

IEC61850 protocol adapter logging:

2018-09-28 06:24:43.599 [osgp-tst-04] [New I/O server boss #9] INFO o.o.a.p.i.i.n.Iec61850
2018-09-28 06:24:43.599 [osgp-tst-04] [New I/O server boss #9] INFO o.o.a.p.i.i.n.Iec61850
2018-09-28 06:24:43.599 [osgp-tst-04] [New I/O server boss #9] INFO o.o.a.p.i.i.n.Iec61850
2018-09-28 06:24:43.599 [osgp-tst-04] [New I/O server boss #9] INFO o.o.j.n.handler.logging.Loggi
2018-09-28 06:24:43.599 [osgp-tst-04] [New I/O server boss #9] INFO o.o.j.n.handler.logging.Loggi
2018-09-28 06:24:43.599 [osgp-tst-04] [New I/O server boss #9] INFO o.o.a.p.i.i.m.OsgpRequestMe
++-----------------------------------------------+
| 0000000053,84.30.69.148 | 0000000053,84.30.69.148 |
++-----------------------------------------------+
| 0000000053,84.30.69.148 | 0000000053,84.30.69.148 |
| 0000000053,84.30.69.148 | 0000000053,84.30.69.148 |

IEC61850 platform messages of the data sent to the device:

LogicalDevice: SWDeviceGenericIO
messageType: LocationInformation {
  CSLC.Atnm[CF].lon: 52.37875
  CSLC.Atnm[CF].lat: 5.95558
}
messageType: DisableRegistration {
  CSLC.Reg[CF].ntEnb: false
}

IEC61850 protocol adapter logging:

2018-09-28 06:24:43.590 [osgp-tst-04] [New I/O server boss #9] INFO o.o.a.p.i.i.n.Iec61850
2018-09-28 06:24:43.590 [osgp-tst-04] [New I/O server boss #9] INFO o.o.a.p.i.i.n.Iec61850
2018-09-28 06:24:43.590 [osgp-tst-04] [New I/O server boss #9] INFO o.o.a.p.i.i.n.Iec61850
2018-09-28 06:24:43.590 [osgp-tst-04] [New I/O server boss #9] INFO o.o.j.n.handler.logging.Loggi
2018-09-28 06:24:43.590 [osgp-tst-04] [New I/O server boss #9] INFO o.o.j.n.handler.logging.Loggi
2018-09-28 06:24:43.590 [osgp-tst-04] [New I/O server boss #9] INFO o.o.a.p.i.i.m.OsgpRequestMe
++-----------------------------------------------+
| 0000000053,84.30.69.148| 0000000053,84.30.69.148 |
++-----------------------------------------------+
| 0000000053,84.30.69.148| 0000000053,84.30.69.148 |
| 0000000053,84.30.69.148| 0000000053,84.30.69.148 |
2018-09-28 06:24:43.599 [osgp-tst-04] [New I/O server boss #9] INFO o.o.j.n.handler.logging.Loggi
2018-09-28 06:24:43.599 [osgp-tst-04] [New I/O server boss #9] INFO o.o.j.n.handler.logging.Loggi
2018-09-28 06:24:43.599 [osgp-tst-04] [New I/O server boss #9] INFO o.o.a.p.i.i.m.OsgpRequestMe
++-----------------------------------------------+
| 0000000053,84.30.69.148| 0000000053,84.30.69.148 |
++-----------------------------------------------+
| 0000000053,84.30.69.148| 0000000053,84.30.69.148 |
| 0000000053,84.30.69.148| 0000000053,84.30.69.148 |
2018-09-28 06:24:43.599 [osgp-tst-04] [New I/O server boss #9] INFO o.o.j.n.handler.logging.Loggi
2018-09-28 06:24:43.599 [osgp-tst-04] [New I/O server boss #9] INFO o.o.j.n.handler.logging.Loggi
2018-09-28 06:24:43.599 [osgp-tst-04] [New I/O server boss #9] INFO o.o.a.p.i.i.m.OsgpRequestMe
++-----------------------------------------------+
| 0000000053,84.30.69.148| 0000000053,84.30.69.148 |
++-----------------------------------------------+
GetConfiguration

GetConfiguration messages

Description

Request which queries a device for its current configuration.

Response which returns the result of the request and, if 'result = OK', contains the configuration of the device.

IEC61850 Fields

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>FC</th>
<th>SUB ATTRIBUTE</th>
<th>DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>XSWC1.SwType</td>
<td>ST</td>
<td>stVal</td>
<td>INT8</td>
<td>Switch type for relay 1, tariff = 0, light = 1.</td>
</tr>
<tr>
<td>XSWC2.SwType</td>
<td>ST</td>
<td>stVal</td>
<td>INT8</td>
<td>Switch type for relay 2, tariff = 0, light = 1.</td>
</tr>
<tr>
<td>XSWC3.SwType</td>
<td>ST</td>
<td>stVal</td>
<td>INT8</td>
<td>Switch type for relay 3, tariff = 0, light = 1.</td>
</tr>
<tr>
<td>CSLC.SWCf</td>
<td>CF</td>
<td>LT</td>
<td>VisString64</td>
<td>Device light type, always set to RELAY.</td>
</tr>
<tr>
<td>CSLC.SWCf</td>
<td>CF</td>
<td>adSetOft</td>
<td>INT16</td>
<td>Offset in minutes with respect to astronomical sunset.</td>
</tr>
<tr>
<td>CSLC.SWCf</td>
<td>CF</td>
<td>adRiseOft</td>
<td>INT16</td>
<td>Offset in minutes with respect to astronomical sunrise.</td>
</tr>
<tr>
<td>CSLC.Reg</td>
<td>CF</td>
<td>svrAddr</td>
<td>VisString64</td>
<td>OSGP server address for device registration.</td>
</tr>
<tr>
<td>CSLC.Reg</td>
<td>CF</td>
<td>svrPort</td>
<td>INT32</td>
<td>OSGP server port for device registration.</td>
</tr>
<tr>
<td>CSLC.Clock</td>
<td>CF</td>
<td>dstBegT</td>
<td>VisString255</td>
<td>Daylight savings time begin time.</td>
</tr>
<tr>
<td>CSLC.Clock</td>
<td>CF</td>
<td>dstEndT</td>
<td>VisString255</td>
<td>Daylight savings time end time.</td>
</tr>
<tr>
<td>CSLC.Clock</td>
<td>CF</td>
<td>enbDst</td>
<td>BOOLEAN</td>
<td>Flag indicating whether daylight savings time is enabled.</td>
</tr>
<tr>
<td>CSLC.Clock</td>
<td>CF</td>
<td>enbNtpC</td>
<td>BOOLEAN</td>
<td>Flag indicating whether NTP client is enabled.</td>
</tr>
<tr>
<td>CSLC.Clock</td>
<td>CF</td>
<td>ntpSrva</td>
<td>VisString255</td>
<td>NTP server address.</td>
</tr>
<tr>
<td>CSLC.Clock</td>
<td>CF</td>
<td>syncPer</td>
<td>INT16U</td>
<td>Time sync period.</td>
</tr>
<tr>
<td>CSLC.IPCf</td>
<td>CF</td>
<td>enbDHCP</td>
<td>BOOLEAN</td>
<td>Flag indicating whether DHCP client is enabled.</td>
</tr>
<tr>
<td>CSLC.IPCf</td>
<td>CF</td>
<td>ipAddr</td>
<td>VisString32</td>
<td>Fixed IP address when DHCP is disabled.</td>
</tr>
<tr>
<td>CSLC.IPCf</td>
<td>CF</td>
<td>netmask</td>
<td>VisString32</td>
<td>Netmask when DHCP is disabled.</td>
</tr>
<tr>
<td>CSLC.IPCf</td>
<td>CF</td>
<td>gateway</td>
<td>VisString32</td>
<td>Gateway when DHCP is disabled.</td>
</tr>
</tbody>
</table>

Example

Soap requests and responses sent to and from platform:

```xml
<soapenv:Envelope
 xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:nm1="http://www.opensmartgridplatform.org/schemas/configurationmanagement/2014/10"
 xmlns:nm2="http://www.opengridplatform.org/schemas/common/2014/10"
 xmlns:nm3="http://www.intelligrated.com/schemas/" >
 <soapenv:Header>
  <ns1:ApplicationName>Sander</ns1:ApplicationName>
  <ns1:UserName>Sander</ns1:UserName>
  <ns1:OrganisationIdentification>LianderNetManagement</ns1:OrganisationIdentification>
 </soapenv:Header>
 <soapenv:Body>
  <ns1:GetConfigurationRequest
   xmlns:nm1="http://www.opensmartgridplatform.org/schemas/configurationmanagement/2014/10"
   xmlns:nm2="http://www.opengridplatform.org/schemas/common/2014/10"
   xmlns:nm3="http://www.intelligrated.com/schemas/" >
   <ns1:DeviceIdentification>
    <nm1:DeviceIdentification>
     <nm2:Name>KAI-0000000053</nm2:Name>
    </ns1:DeviceIdentification>
   </ns1:GetConfigurationRequest>
  </soapenv:Body>
</soapenv:Envelope>

<SOAP-ENV:Envelope
 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
 <SOAP-ENV:Header/>
 <SOAP-ENV:Body>
  <ns2:GetConfigurationAsyncResponse
   xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:nm2="http://www.opengridplatform.org/schemas/configurationmanagement/2014/10"
   xmlns:nm3="http://www.intelligrated.com/schemas/" >
   <ns2:AsyncResponse>
    <ns3:CorrelationUid>LianderNetManagement|||KAI-0000000053|||20180925073838432<,
    <ns3:DeviceId>KAI-0000000053</ns3:DeviceId>
   </ns2:AsyncResponse>
  </ns2:GetConfigurationAsyncResponse>
</SOAP-ENV:Body>
```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  xmlns:ns1="http://www.opensmartgridplatform.org/schemas/common/2014/10"
  xmlns:ns2="http://www.opensmartgridplatform.org/schemas/configurationmanagement/2014/10"
  xmlns:ns3="http://www.opensmartgridplatform.org/schemas/common/2014/10"
  xmlns:ns="http://www.opensmartgridplatform.org/schemas/configurationmanagement/2014/10"

  <soapenv:Header>
    <ns:ApplicationName>SoapUI</ns:ApplicationName>
    <ns:UserName>Sander</ns:UserName>
    <ns:OrganisationIdentification>LianderNetManagement</ns:OrganisationIdentification>
  </soapenv:Header>

  <soapenv:Body>
    <ns2:GetConfigurationAsyncRequest>
      <ns:AsyncRequest>
        <ns:CorrelationUid>LianderNetManagement|||KAI-0000000053|||20180925073838432</ns:CorrelationUid>
        <ns:DeviceId>KAI-0000000053</ns:DeviceId>
      </ns:AsyncRequest>
    </ns2:GetConfigurationAsyncRequest>
  </soapenv:Body>

</soapenv:Envelope>

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  xmlns:ns="http://www.opensmartgridplatform.org/schemas/configurationmanagement/2014/10"
  xmlns:ns2="http://www.opensmartgridplatform.org/schemas/configurationmanagement/2014/10"
  xmlns:ns3="http://www.opensmartgridplatform.org/schemas/common/2014/10"

  <soapenv:Header>
    <ns:Result>OK</ns:Result>
  </soapenv:Header>

  <soapenv:Body>
    <ns2:GetConfigurationResponse>
      <ns2:Configuration>
        <ns2:LightType>RELAY</ns2:LightType>
        <ns2:RelayConfiguration>
          <ns2:RelayMap>
            <ns2:Index>1</ns2:Index>
            <ns2:Address>1</ns2:Address>
            <ns2:RelayType>TARIFF</ns2:RelayType>
            <ns2:alias/>
          </ns2:RelayMap>
          <ns2:RelayMap>
            <ns2:Index>2</ns2:Index>
            <ns2:Address>2</ns2:Address>
            <ns2:RelayType>LIGHT</ns2:RelayType>
            <ns2:alias/>
          </ns2:RelayMap>
          <ns2:RelayMap>
            <ns2:Index>3</ns2:Index>
            <ns2:Address>3</ns2:Address>
            <ns2:RelayType>LIGHT</ns2:RelayType>
            <ns2:alias/>
          </ns2:RelayMap>
        </ns2:RelayConfiguration>
        <ns2:PreferredLinkType>ETHERNET</ns2:PreferredLinkType>
        <ns2:TimeSyncFrequency>1440</ns2:TimeSyncFrequency>
        <ns2:DeviceFixedIp>
          <ns2:IpAddress>192.168.0.110</ns2:IpAddress>
          <ns2:NetMask>255.255.0.0</ns2:NetMask>
          <ns2:GateWay>192.168.0.1</ns2:GateWay>
        </ns2:DeviceFixedIp>
        <ns2:DhcpEnabled>true</ns2:DhcpEnabled>
        <ns2:OsgpIpAddress>168.63.97.65</ns2:OsgpIpAddress>
        <ns2:OsgpPortNumber>50003</ns2:OsgpPortNumber>
        <ns2:NtpHost>0.nl.pool.ntp.org</ns2:NtpHost>
        <ns2:NtpEnabled>true</ns2:NtpEnabled>
        <ns2:NtpSyncInterval>1440</ns2:NtpSyncInterval>
        <ns2:AutomaticSummerTimingEnabled>true</ns2:AutomaticSummerTimingEnabled>
        <ns2:AstroGateSunRiseOffset>0</ns2:AstroGateSunRiseOffset>
        <ns2:AstroGateSunSetOffset>0</ns2:AstroGateSunSetOffset>
        <ns2:SummerTimeDetails>2019-03-30T23:00:00.000Z</ns2:SummerTimeDetails>
        <ns2:WinterTimeDetails>2018-10-27T22:00:00.000Z</ns2:WinterTimeDetails>
      </ns2:Configuration>
    </ns2:GetConfigurationResponse>
  </soapenv:Body>
</soapenv:Envelope>
Platform message of the data read from the device:

**LogicalDevice**: SWDeviceGenericIO

**messageType**: GetConfiguration {

- XSWC1.SwType[ST].stVal: 0
- XSWC2.SwType[ST].stVal: 1
- XSWC3.SwType[ST].stVal: 1
- CSLC.SWCf[CF].LT: RELAY
- CSLC.SWCf[CF].adSetOft: 0
- CSLC.SWCf[CF].adRiseOft: 0
- CSLC.Reg[CF].svrPort: 50003
- CSLC.Reg[CF].svrAddr: 168.63.97.65
- CSLC.Clock[CF].dstBegT: M3.5.0
- CSLC.Clock[CF].dstEndT: M10.5.0
- CSLC.Clock[CF].enbDst: true
- CSLC.Clock[CF].enbNtpC: true
- CSLC.Clock[CF].ntpSvrA: 0.nl.pool.ntp.org
- CSLC.Clock[CF].syncPer: 1440
- CSLC.IPCf[CF].enbDHCP: true
- CSLC.IPCf[CF].ipAddr: 192.168.0.110
- CSLC.IPCf[CF].netmask: 255.255.0.0
- CSLC.IPCf[CF].gateway: 192.168.0.1

**IEC61850** protocol adapter logging:

```
2018-09-25 07:38:39.074] [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-25 07:38:39.091] [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-25 07:38:39.146] [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-25 07:38:39.163] [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
```
SetConfiguration

SetConfiguration messages

**Description**

Request which commands a device to update its configuration.

Response which returns the result of the request.

**IEC61850 Fields**

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>FC SUB ATTRIBUTE DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>XSWC1.SwType CO Oper.ctlVal</td>
<td>INT8</td>
<td>Switch type for relay 1, tariff = 0, light = 1.</td>
</tr>
<tr>
<td>XSWC2.SwType CO Oper.ctlVal</td>
<td>INT8</td>
<td>Switch type for relay 2, tariff = 0, light = 1.</td>
</tr>
<tr>
<td>XSWC3.SwType CO Oper.ctlVal</td>
<td>INT8</td>
<td>Switch type for relay 3, tariff = 0, light = 1.</td>
</tr>
<tr>
<td>XSWC4.SwType CO Oper.ctlVal</td>
<td>INT8</td>
<td>Switch type for relay 4, tariff = 0, light = 1.</td>
</tr>
<tr>
<td>CSLC.SWCf CF LT VisString64</td>
<td>Light type, always set to &quot;RELAY&quot;.</td>
<td></td>
</tr>
<tr>
<td>CSLC.SWCf CF adSetOft INT16</td>
<td>Offset in minutes with respect to astronomical sunset.</td>
<td></td>
</tr>
<tr>
<td>CSLC.SWCf CF adRiseOft INT16</td>
<td>Offset in minutes with respect to astronomical sunrise.</td>
<td></td>
</tr>
<tr>
<td>CSLC.Reg CF svrAddr VisString64 OSGP</td>
<td>OSGP server address for device registration.</td>
<td></td>
</tr>
<tr>
<td>CSLC.Reg CF svrPort INT32 OSGP</td>
<td>OSGP server port for device registration.</td>
<td></td>
</tr>
<tr>
<td>CSLC.Clock CF dstBegT VisString255 Daylight savings time begin time.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSLC.Clock CF dstEndT VisString255 Daylight savings time end time.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSLC.Clock CF enbDst BOOLEAN Flag indicating whether daylight savings time is enabled.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSLC.Clock CF enbNtpC BOOLEAN Flag indicating whether NTP client is enabled.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSLC.Clock CF ntpSvrA VisString255 NTP server address.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSLC.Clock CF syncPer INT16U</td>
<td>Time sync period in minutes.</td>
<td></td>
</tr>
<tr>
<td>CSLC.IPCf CF enbDHCP BOOLEAN Flag indicating whether DHCP client is enabled.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSLC.IPCf CF ipAddr VisString32 Fixed IP address when DHCP is disabled.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSLC.IPCf CF netmask VisString32 Netmask when DHCP is disabled.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSLC.IPCf CF gateway VisString32 Gateway when DHCP is disabled.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Example**

Soap requests and responses sent to and from platform:

```xml
<soapenv:Envelope
 xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
 xmlns:ns="http://www.opensmartgridplatform.org/schemas/common/2014/10"
 xmlns:ns1="http://www.opensmartgridplatform.org/schemas/configurationmanagement/2014/10"

<soapenv:Header>
 <ns:ApplicationName>SoapUI</ns:ApplicationName>
 <ns:UserName>Sander</ns:UserName>
 <ns:OrganisationIdentification>LianderNetManagement</ns:OrganisationIdentification>
</soapenv:Header>

<soapenv:Body>
 <ns1:SetConfigurationRequest>
  <ns1:DeviceIdentification>KAI-0000000053</ns1:DeviceIdentification>
  <ns1:Configuration>
   <ns1:LightType>RELAY</ns1:LightType>
   <ns1:RelayConfiguration>
    <ns1:RelayMap>
     <ns1:Index>1</ns1:Index>
     <ns1:Address>1</ns1:Address>
     <ns1:RelayType>TARIFF</ns1:RelayType>
    </ns1:RelayMap>
    <ns1:RelayMap>
     <ns1:Index>2</ns1:Index>
     <ns1:Address>2</ns1:Address>
     <ns1:RelayType>LIGHT</ns1:RelayType>
    </ns1:RelayMap>
   </ns1:RelayConfiguration>
  </ns1:Configuration>
</soapenv:Body>
```

SWDevice-010805 389
<ns1:RelayType>LIGHT</ns1:RelayType>
</ns1:RelayConfiguration>
<ns1:PreferredLinkType>ETHERNET</ns1:PreferredLinkType>
<ns1:TimeSyncFrequency>1440</ns1:TimeSyncFrequency>
<ns1:DeviceFixedIp>
<ns1:IpAddress>192.168.0.110</ns1:IpAddress>
<ns1:NetMask>255.255.0.0</ns1:NetMask>
<ns1:GateWay>192.168.0.1</ns1:GateWay>
</ns1:DeviceFixedIp>
<ns1:DhcpEnabled>true</ns1:DhcpEnabled>
<ns1:OsgpIpAddress>168.63.97.65</ns1:OsgpIpAddress>
<ns1:OsgpPortNumber>50003</ns1:OsgpPortNumber>
<ns1:NtpHost>0.nl.pool.ntp.org</ns1:NtpHost>
<ns1:NtpEnabled>true</ns1:NtpEnabled>
<ns1:NtpSyncInterval>1440</ns1:NtpSyncInterval>
<ns1:AutomaticSummerTimingEnabled>true</ns1:AutomaticSummerTimingEnabled>
<ns1:AstroGateSunRiseOffset>0</ns1:AstroGateSunRiseOffset>
<ns1:AstroGateSunSetOffset>0</ns1:AstroGateSunSetOffset>
<ns1:SummerTimeDetails>2019-03-30T23:00:00.000Z</ns1:SummerTimeDetails>
<ns1:WinterTimeDetails>2018-10-27T22:00:00.000Z</ns1:WinterTimeDetails>
</ns1:Configuration>
</soapenv:Body>
</SOAP-ENV:Envelope>

<SOAP-ENV:Header>
<ns2:AsyncResponse>
<ns3:CorrelationUid>LianderNetManagement|||KAI-0000000053|||20180925135306803</ns3:CorrelationUid>
<ns3:DeviceId>KAI-0000000053</ns3:DeviceId>
</ns2:AsyncResponse>
</ns2:SetConfigurationAsyncResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

<SOAP-ENV:Header>
<ns:ApplicationName>SoapUI</ns:ApplicationName>
<ns:UserName>Sander</ns:UserName>
<ns:OrganisationIdentification>Liander NetManagement</ns:OrganisationIdentification>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
<ns1:SetConfigurationAsyncRequest>
<ns1:AsyncRequest>
<ns1:CorrelationUid>LianderNetManagement|||KAI-0000000053|||20180925135306803</ns1:CorrelationUid>
<ns1:DeviceId>KAI-0000000053</ns1:DeviceId>
</ns1:AsyncRequest>
</ns1:SetConfigurationAsyncRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

<SOAP-ENV:Header>
<ns2:SetConfigurationResponse xmlns:ns2="http://www.opensmartgridplatform.org/schemas/configurationmanagement/2014/10">
<ns2:Result>OK</ns2:Result>
</ns2:SetConfigurationResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
IEC61850 platform message of the data sent to the device:

LogicalDevice: SWDeviceGenericIO
messageType: SetConfiguration

```
  CSLC.Reg[CF].svrPort: 50003
  CSLC.SWCf[CF].adSetOft: 0
  CSLC.Clock[CF].ntpSvrA: 0.nl.pool.ntp.org
  CSLC.SWCf[CF].LT: RELAY
  CSLC.Clock[CF].dstEndT: M10.5.6/22
  CSLC.IPCf[CF].enbDHCP: true
  XSWC2.SwType[CO].Oper.ctlVal: 1
  XSWC3.SwType[CO].Oper.ctlVal: 1
  CSLC.Clock[CF].enbDst: true
  XSWC1.SwType[CO].Oper.ctlVal: 0
  CSLC.Clock[CF].enbNtpC: true
  CSL.IPCF[CF].netmask: 255.255.0.0
  CSL.IPCF[CF].gateway: 192.168.0.1
  CSL.IPCF[CF].ipAddr: 192.168.0.110
  CSLC.Reg[CF].svrAddr: 168.63.97.65
  CSLC.SWCf[CF].adRiseOft: 0
  CSLC.Clock[CF].syncPer: 1440
  CSLC.Clock[CF].dstBegT: M3.5.6/23
```
SetEventNotifications

SetEventNotifications messages

Description
Request which commands a device to set its EventNotification mask.

Response which returns the result of the request.

Message definitions

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>SUB Attribute</th>
<th>DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSLC.EvnBuf</td>
<td>CF enbEvnType</td>
<td>VisString32</td>
<td>Bitmask indicating which event notification types are enabled, to enable all event types use: &quot;3FFFFFF&quot;.</td>
</tr>
</tbody>
</table>

Example
Soap requests and responses sent to and from platform:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:ns="http://www.opensmartgridplatform.org/schemas/common/2014/10"
    xmlns:ns1="http://www.opensmartgridplatform.org/schemas/devicemanagement/2014/10">
  <soapenv:Header>
    <ns:ApplicationName>SoapUI</ns:ApplicationName>
    <ns:UserName>Sander</ns:UserName>
    <ns:OrganisationIdentification>LianderNetManagement</ns:OrganisationIdentification>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:SetEventNotificationsRequest>
      <ns1:DeviceIdentification>KAI-0000000053</ns1:DeviceIdentification>
      <ns1:EventNotifications>
        <ns1:EVENT>DIAG_EVENTS</ns1:EventNotifications>
        <ns1:EventNotifications>HARDWARE_FAILURE</ns1:EventNotifications>
        <ns1:EventNotifications>LIGHT_EVENTS</ns1:EventNotifications>
        <ns1:EventNotifications>TARIFF_EVENTS</ns1:EventNotifications>
        <ns1:EventNotifications>MONITOR_EVENTS</ns1:EventNotifications>
        <ns1:EventNotifications>COMM_EVENTS</ns1:EventNotifications>
        <ns1:EventNotifications>SECURITY_EVENTS</ns1:EventNotifications>
      </ns1:EventNotifications>
    </ns1:SetEventNotificationsRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

```xml
  xmlns:ns2="http://www.opensmartgridplatform.org/schemas/devicemanagement/2014/10"
  xmlns:ns3="http://www.opensmartgridplatform.org/schemas/common/2014/10">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:SetEventNotificationsAsyncResponse>
      <ns:CorrelationUid>LianderNetManagement|||KAI-0000000053|||20180927074508777</ns:CorrelationUid>
      <ns3:DeviceId>KAI-0000000053</ns3:DeviceId>
    </ns2:SetEventNotificationsAsyncResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
Platform message of data written to the device:

LogicalDevice: SWDeviceGenericIO
messageType: SetEventNotificationFilter {
  CSLC.EvnBuf[CF].enbEvnType: 3FFFFFF
}

IEC61850 protocol adapter logging:

2018-09-27 07:45:09.181 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-27 07:45:09.225 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-27 07:45:09.225 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-27 07:45:09.225 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-27 07:45:09.231 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-27 07:45:09.233 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-27 07:45:09.282 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-27 07:45:09.282 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-27 07:45:09.282 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-27 07:45:09.313 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-27 07:45:09.313 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-27 07:45:09.329 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-27 07:45:09.329 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-27 07:45:09.348 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-27 07:45:09.349 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-27 07:45:09.349 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-27 07:45:09.349 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-27 07:45:09.349 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-27 07:45:09.349 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-27 07:45:09.349 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-27 07:45:09.349 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-27 07:45:09.349 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-27 07:45:09.349 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-27 07:45:09.349 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-27 07:45:09.349 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-27 07:45:09.349 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-27 07:45:09.349 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-27 07:45:09.349 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-27 07:45:09.349 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o

**EventNotification**

**EventNotification messages**

**Description**

Buffered report sent from device to platform containing information about 1 event. The devices keep up to 120 events in the cyclic buffer CSLC.EvnBuf.evnl/env120. The events are converted to buffered reports when OSGP triggers the device to do so by writing true to CSLC.evn_rpn01[BR].RptEna. When this happens, the device will send the buffered reports to OSGP. OSGP will save the information specified by the remark field, like the event type and the event time:

- **evnType**: 4 = TARIFF_EVENTS_TARIFF_ON
- **swNum**: 1 = get external index for switch 1
- **trgType**: 3 = fixed time trigger
- **swVal**: true = ON
- **trgTime**: 2018-10-01T05:00:00.000Z

Example buffered report:

```plaintext
RptId: evn_rpn
DataSetRef: SWDeviceGenericIO/LLN0.evn_rpn
ConfRev: null
BufOvfl: true
EntryId: none: [0, 0, 1, 102, 45, -1, 87, -73] (f-wO)
InclusionBitString: [true]
MoreSegmentsFollow: false
SqNum: 0
SubSqNum: null
TimeOfEntry: none: 1096606800823 (2018-10-01T05:00:00.823Z)
ReasonCodes: 0x40 (DataChange)
DataSet: SWDeviceGenericIO/LLN0.evn_rpn
DataSet members: 1
member: SWDeviceGenericIO/CSLC.EvnRpn [ST]

SwDeviceGenericIO/CSLC.EvnRpn.evnType: 4
SwDeviceGenericIO/CSLC.EvnRpn.swNum: 1
SwDeviceGenericIO/CSLC.EvnRpn.trgType: 3
SwDeviceGenericIO/CSLC.EvnRpn.swVal: true
SwDeviceGenericIO/CSLC.EvnRpn.trgTime: Mon Oct 01 05:00:00 UTC 2018
SwDeviceGenericIO/CSLC.EvnRpn.remark: remark

NOTE: After executing operations SetLightRequest, GetStatusRequest and SetTransitionRequest, OSGP will enable reporting on the device. Optionally, OSGP will enable reporting after DeviceRegistrationRequest (default is false).

**IEC61850 Fields**

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>FC</th>
<th>SUB ATTRIBUTE</th>
<th>DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSLC.evn_rpn01</td>
<td>BR</td>
<td>RptEna</td>
<td>BOOLEAN</td>
<td>Flag which indicates to the device to send buffered reports.</td>
</tr>
</tbody>
</table>

**IEC61850** platform message of the data sent to the device:

```plaintext
LogicalDevice: SWDeviceGenericIO
messageType: EnableBufferedReporting {
  CSLC.evn_rpn01[BR].RptEna: true
}
```

**IEC61850** protocol Adapter logging:

- 2018-10-01 18:52:10.476 [osgp-tst-04] [iec61850RequestsMessageListenerContainer-7] INFO ...
- 2018-10-01 18:52:10.477 [osgp-tst-04] [iec61850RequestsMessageListenerContainer-7] INFO ...
- 2018-10-01 18:52:10.477 [osgp-tst-04] [iec61850RequestsMessageListenerContainer-7] INFO ...
- 2018-10-01 18:52:10.480 [osgp-tst-04] [iec61850RequestsMessageListenerContainer-7] INFO ...

SWDevice-010805 394

2018-10-01 10:52:14.796 [osgp-tst-04] [Thread-327] INFO o.o.a.p.i.i.n.r.Iec61850ClientSSLDEventListener@logReportDetails:273 - Report details for device KAI-0000000053

2018-10-01 10:52:14.058 [osgp-tst-04] [iec61850RequestsMessageListenerContainer-7] INFO o.o.a.p.i.i.n.h.DeviceConnection@createObjectReference:94 - Device: KAI-0000000053, ObjectReference: SWDeviceGenericIO/LLN0.evn_rpn

2018-10-01 10:52:14.058 [osgp-tst-04] [iec61850RequestsMessageListenerContainer-7] INFO o.o.a.p.i.i.n.helper.NodeContainer@getString:88 - Device: KAI-0000000053, enbEvnType has value 1FFFFFF

2018-10-01 10:52:14.058 [osgp-tst-04] [iec61850RequestsMessageListenerContainer-7] INFO o.o.a.p.i.i.n.s.c.Iec61850GetStatusCommand@apply:63 - Got status of relay 1 => on

2018-10-01 10:52:14.058 [osgp-tst-04] [iec61850RequestsMessageListenerContainer-7] INFO o.o.a.p.i.i.n.s.c.Iec61850GetStatusCommand@apply:74 - Got EvnBuf.enbEvnType filter 1FFFFFF

2018-10-01 10:52:14.064 [osgp-tst-04] [iec61850RequestsMessageListenerContainer-7] INFO o.o.a.p.i.i.n.s.c.Iec61850GetStatusCommand@apply:63 - Got status of relay 2 => off


2018-10-01 10:52:14.058 [osgp-tst-04] [iec61850RequestsMessageListenerContainer-7] INFO o.o.a.p.i.i.n.helper.NodeContainer@getString:88 - Device: KAI-0000000053, enbEvnType has value 1FFFFFF

ReasonCodes: 0x40 (DataChange)

DataSet: SWDeviceGenericIO/LLN0.evn_rpn
DataSet members: 1

Member: SWDeviceGenericIO/CSLC.EvnRpn [ST]

SWDeviceGenericIO/CSLC.EvnRpn.evtType: 3
SWDeviceGenericIO/CSLC.EvnRpn.swNum: 2
SWDeviceGenericIO/CSLC.EvnRpn.trgType: 1
SWDeviceGenericIO/CSLC.EvnRpn.swVal: false
SWDeviceGenericIO/CSLC.EvnRpn.trgTime: Mon Oct 01 05:51:01 UTC 2018
SWDeviceGenericIO/CSLC.EvnRpn.remark: remark

evtType: 3 = LIGHT_EVENTS_LIGHT_OFF
swNum: 2 = get external index for switch 2
trgType: 1 = light trigger (sensor trigger)
swVal: false = OFF
trgTime: 2018-10-01T05:51:01.000Z
remark: remark

2018-10-01 10:52:14.799] [osgp-tst-04] [Thread-328] INFO o.o.a.p.i.i.n.r.Iec61850ClientSSLData
2018-10-01 10:52:19.510] [osgp-tst-04] [Timer-52] INFO o.o.a.p.i.i.n.h.DeviceConnection@createObjectReference:94 - Device: KAI-0000000053, ObjectReference: SWDeviceGenericIO/LLN0.evn_rpn01
2018-10-01 10:52:21.301] [osgp-tst-04] [Thread-329] INFO o.o.a.p.i.i.n.r.Iec61850ClientSSLData@associationClosed:366 - associationClosed() for device: KAI-0000000053, IOException: Connection disconnected by client
2018-10-01 10:52:21.304] [osgp-tst-04] [Thread-329] INFO o.o.a.p.i.i.m.OsgpRequestMessageSender@send:34 - Sending request message to OSGP.
SetSchedule

SetSchedule messages

Description
Request which commands a device to set a light or tariff schedule.

Response which returns the result of the request.

IEC61850 Fields

The table shows the fields for XSWC1 (relay 1). The device has 4 relays (XSWC1...XSWC4).

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>SUB ATTRIBUTE</th>
<th>DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>XSWC1.Sche</td>
<td>CF sche1.enable</td>
<td>BOOLEAN</td>
<td>Flag indicating the schedule entry is enabled.</td>
</tr>
<tr>
<td>XSWC1.Sche</td>
<td>CF sche1.day</td>
<td>INT32</td>
<td>Day in yyyyymmdd format or defined by DAY enum.</td>
</tr>
<tr>
<td>XSWC1.Sche</td>
<td>CF sche1.tOn</td>
<td>INT32</td>
<td>Timestamp in hhmm format when relay should switch on or -1 if not used.</td>
</tr>
<tr>
<td>XSWC1.Sche</td>
<td>CF sche1.tOnT</td>
<td>INT8</td>
<td>Schedule entry type, 0 = fixed time, 1 = light sensor, 2 = astronomical time.</td>
</tr>
<tr>
<td>XSWC1.Sche</td>
<td>CF sche1.tOff</td>
<td>INT32</td>
<td>Timestamp in hhmm format when relay should switch off or -1 if not used.</td>
</tr>
<tr>
<td>XSWC1.Sche</td>
<td>CF sche1.tOffT</td>
<td>INT8</td>
<td>Schedule entry type, 0 = fixed time, 1 = light sensor, 2 = astronomical time.</td>
</tr>
<tr>
<td>XSWC1.Sche</td>
<td>CF sche1.minOnPer</td>
<td>INT16U</td>
<td>Minimum burning time for this relay.</td>
</tr>
<tr>
<td>XSWC1.Sche</td>
<td>CF sche1.minOffPer</td>
<td>INT16U</td>
<td>Not used.</td>
</tr>
<tr>
<td>XSWC1.Sche</td>
<td>CF sche1.srBefWd</td>
<td>INT16U</td>
<td>Window for light sensor trigger, minutes before astronomical time.</td>
</tr>
<tr>
<td>XSWC1.Sche</td>
<td>CF sche1.srAftWd</td>
<td>INT16U</td>
<td>Window for light sensor trigger, minutes after astronomical time.</td>
</tr>
<tr>
<td>XSWC1.Sche</td>
<td>CF sche1.igBefWd</td>
<td>INT16U</td>
<td>Not used.</td>
</tr>
<tr>
<td>XSWC1.Sche</td>
<td>CF sche1.igAftWd</td>
<td>INT16U</td>
<td>Not used.</td>
</tr>
</tbody>
</table>

Although the device supports setting 64 schedule entries (sche1...sche64) for 4 relays (XSWC1...XSWC4), the actual number of schedule entries is limited by OSGP to 50.

Besides the fields on the relay, the switch logic applies astronomical sunrise and sunset offsets to the calculated astronomical times. These are stored with the Street Light Configuration (logical node CSLC).

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>SUB ATTRIBUTE</th>
<th>DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWCf</td>
<td>adSetOft</td>
<td>INT16</td>
<td>Offset to be used with calculated astronomical sunset time.</td>
</tr>
<tr>
<td>SWCf</td>
<td>adRiseOft</td>
<td>INT16</td>
<td>Offset to be used with calculated astronomical sunrise time.</td>
</tr>
</tbody>
</table>

enum DAY {
    0 Every day of the week;
    -1 Every weekday: Monday, Tuesday, Wednesday, Thursday, Friday;
    -2 Every weekend day: Saturday, Sunday;
    1 Monday;
    2 Tuesday;
    3 Wednesday;
    4 Thursday;
    5 Friday;
    6 Saturday;
    7 Sunday;
}

Examples

Example 1: Light schedule based on light measurement

Description: This schedule combines a 'morning/evening light' with an 'all night light'. Relay 1 and 2 will be switched on using a light measurement trigger. Relay 2 will be switched off at 23:00 using an absolute time. Relay 2 will be switched on at 07:00, but only when no light measurement trigger has been received yet. Relay 1 and 2 will be switched off using a light measurement trigger.

Screenshot of this schedule in an OSGP client application:
Soap Request Message for Platform web service:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:ns="http://www.opensmartgridplatform.org/schemas/common/2014/10"
                  xmlns:ns1="http://www.opensmartgridplatform.org/schemas/publiclighting/schedulemanagement"

  <ns:ApplicationName>SoapUI</ns:ApplicationName>
  <ns:UserName>Sander</ns:UserName>
  <ns:OrganisationIdentification>LianderNetManagement</ns:OrganisationIdentification>
</soapenv:Header>

<soapenv:Body>
  <ns1:SetScheduleRequest>
    <ns1:DeviceIdentification>KA1-0000000053</ns1:DeviceIdentification>

    <ns1:Schedules>
      <ns1:Schedule>
        <ns1:DayOfWeek>ALL</ns1:DayOfWeek>
        <ns1:ActionTime>SUNRISE</ns1:ActionTime>
        <ns1:TriggerWindow>
          <ns1:MinutesBefore>15</ns1:MinutesBefore>
          <ns1:MinutesAfter>15</ns1:MinutesAfter>
        </ns1:TriggerWindow>
        <ns1:LightValue>
          <ns1:Index>0</ns1:Index>
          <ns1:On>false</ns1:On>
        </ns1:LightValue>
        <ns1:TriggerType>LIGHT_TRIGGER</ns1:TriggerType>
      </ns1:Schedules>

      <ns1:Schedules>
        <ns1:DayOfWeek>ALL</ns1:DayOfWeek>
        <ns1:ActionTime>SUNSET</ns1:ActionTime>
        <ns1:TriggerWindow>
          <ns1:MinutesBefore>15</ns1:MinutesBefore>
          <ns1:MinutesAfter>15</ns1:MinutesAfter>
        </ns1:TriggerWindow>
        <ns1:LightValue>
          <ns1:Index>0</ns1:Index>
          <ns1:On>false</ns1:On>
        </ns1:LightValue>
      </ns1:Schedules>
    </ns1:Schedules>
  </ns1:SetScheduleRequest>
</soapenv:Body>
```
<SOAP-ENV:Body>
  <ns2:SetScheduleResponse
    xmlns:ns2="http://www.opensmartgridplatform.org/schemas/publiclighting/schedulemanagement/2014/10"
    xmlns:ns3="http://www.opensmartgridplatform.org/schemas/common/2014/10">
    <ns2:Result>OK</ns2:Result>
  </ns2:SetScheduleResponse>
</SOAP-ENV:Body>

Platform message of data written to the device:

LogicalDevice: SWDeviceGenericIO
messageType: SetSchedule {
  XSWC2.Sche[CF].sche1.enable: true
  XSWC2.Sche[CF].sche1.tOn: -1
  XSWC2.Sche[CF].sche1.tOnT: -1
  XSWC2.Sche[CF].sche1.tOff: 0
  XSWC2.Sche[CF].sche1.tOffT: 1
  XSWC2.Sche[CF].sche1.srBefWd: 15
  XSWC2.Sche[CF].sche1.srAftWd: 15
  XSWC2.Sche[CF].sche2.enable: true
  XSWC2.Sche[CF].sche2.tOn: 0
  XSWC2.Sche[CF].sche2.tOnT: 1
  XSWC2.Sche[CF].sche2.tOff: -1
  XSWC2.Sche[CF].sche2.tOffT: -1
  XSWC2.Sche[CF].sche2.srBefWd: 15
  XSWC2.Sche[CF].sche2.srAftWd: 15
  XSWC2.Sche[CF].sche3.enable: true
  XSWC2.Sche[CF].sche3.tOn: -1
  XSWC2.Sche[CF].sche3.tOnT: -1
  XSWC2.Sche[CF].sche3.tOff: 2300
  XSWC2.Sche[CF].sche3.tOffT: 0
  XSWC2.Sche[CF].sche3.srBefWd: 30
  XSWC2.Sche[CF].sche3.srAftWd: 30
  XSWC2.Sche[CF].sche4.enable: true
  XSWC2.Sche[CF].sche4.tOn: 700
  XSWC2.Sche[CF].sche4.tOnT: 0
  XSWC2.Sche[CF].sche4.tOff: -1
  XSWC2.Sche[CF].sche4.tOffT: -1
  XSWC2.Sche[CF].sche4.srBefWd: 150
  XSWC2.Sche[CF].sche4.srAftWd: 45
  XSWC3.Sche[CF].sche1.enable: true
  XSWC3.Sche[CF].sche1.tOn: -1
  XSWC3.Sche[CF].sche1.tOnT: -1
  XSWC3.Sche[CF].sche1.tOff: 0
  XSWC3.Sche[CF].sche1.tOffT: 1
  XSWC3.Sche[CF].sche1.srBefWd: 15
  XSWC3.Sche[CF].sche1.srAftWd: 15
  XSWC3.Sche[CF].sche2.enable: true
  XSWC3.Sche[CF].sche2.tOn: 0
  XSWC3.Sche[CF].sche2.tOnT: 1
  XSWC3.Sche[CF].sche2.tOff: -1
  XSWC3.Sche[CF].sche2.tOffT: -1
  XSWC3.Sche[CF].sche2.srBefWd: 15
  XSWC3.Sche[CF].sche2.srAftWd: 15
  XSWC3.Sche[CF].sche2.enable: true
  XSWC3.Sche[CF].sche2.tOn: 0
  XSWC3.Sche[CF].sche2.tOnT: 1
  XSWC3.Sche[CF].sche2.tOff: -1
  XSWC3.Sche[CF].sche2.tOffT: -1
  XSWC3.Sche[CF].sche2.srBefWd: 15
  XSWC3.Sche[CF].sche2.srAftWd: 15
}

IEC61850 protocol adapter logging:

2018-09-26 11:12:39.470] [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-26 11:12:39.471] [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-26 11:12:39.471] [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-26 11:12:39.471] [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-26 11:12:39.475] [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-26 11:12:39.480] [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o

SWDevice-010805 400
Astronomical Offsets

The SOAP request message may contain information about astronomical offsets (see the documentation about light schedules for more details about the offsets).

When AstronomicalSunriseOffset and/or AstronomicalSunsetOffset are set, they will be written to the device in attributes CSLC.SWCF.adRiseoff and CSLC.SWCF.adSetoff.

Example 2: Tariff Schedule

Description for this schedule:

This schedule defines the tariff switching moments. For most weekdays of the year the tariff is high from 7 o'clock in the morning until 11 o'clock in the evening. During the night and weekend, the tariff is low. However for certain days, like Christmas Day, the tariff has to be low as well (Christmas Day may be a weekday).

SOAP Request Message for Platform web service:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:ns1="http://www.opensmartgridplatform.org/schemas/common/2014/10"
                  xmlns:ns="http://www.opensmartgridplatform.org/schemas/common/2014/10">
  <soapenv:Header/>
  <ns1:ApplicationName>SoapUI</ns1:ApplicationName>
  <ns1:UserName>Sander</ns1:UserName>
  <ns1:OrganisationIdentification>LianderNetManagement</ns1:OrganisationIdentification>
</soapenv:Header>

<soapenv:Body>
  <ns1:SetScheduleRequest>
    <ns1:DeviceIdentification>KAI-0000000053</ns1:DeviceIdentification>
    <ns1:Schedules>
      <ns1:WeekDay>WEEKDAY</ns1:WeekDay>
      <ns1:StartDay>2018-01-01</ns1:StartDay>
      <ns1:EndDay>2018-01-01</ns1:EndDay>
      <ns1:StartDay>2018-02-05</ns1:StartDay>
      <ns1:EndDay>2018-02-05</ns1:EndDay>
      <ns1:StartDay>2018-03-05</ns1:StartDay>
      <ns1:EndDay>2018-03-05</ns1:EndDay>
      <ns1:StartDay>2018-04-05</ns1:StartDay>
      <ns1:EndDay>2018-04-05</ns1:EndDay>
      <ns1:StartDay>2018-05-05</ns1:StartDay>
      <ns1:EndDay>2018-05-05</ns1:EndDay>
      <ns1:StartDay>2018-06-05</ns1:StartDay>
      <ns1:EndDay>2018-06-05</ns1:EndDay>
      <ns1:StartDay>2018-07-05</ns1:StartDay>
      <ns1:EndDay>2018-07-05</ns1:EndDay>
      <ns1:StartDay>2018-08-05</ns1:StartDay>
      <ns1:EndDay>2018-08-05</ns1:EndDay>
      <ns1:StartDay>2018-09-05</ns1:StartDay>
      <ns1:EndDay>2018-09-05</ns1:EndDay>
      <ns1:StartDay>2018-10-05</ns1:StartDay>
      <ns1:EndDay>2018-10-05</ns1:EndDay>
      <ns1:StartDay>2018-11-05</ns1:StartDay>
      <ns1:EndDay>2018-11-05</ns1:EndDay>
      <ns1:StartDay>2018-12-05</ns1:StartDay>
      <ns1:EndDay>2018-12-05</ns1:EndDay>
    </ns1:Schedules>
  </ns1:SetScheduleRequest>
</soapenv:Body>
</soapenv:Envelope>
```
<ns1:EndDay>2019-02-01</ns1:EndDay>
<ns1:Time>23:00:00</ns1:Time>
<ns1:TariffValue>
  <ns1:Index>1</ns1:Index>
  <ns1:High>0</ns1:High>
</ns1:TariffValue>
</ns1:Schedules>

<ns1:Schedules>
  <ns1:WeekDay>WEEKDAY</ns1:WeekDay>
  <ns1:StartDay>2018-01-01</ns1:StartDay>
  <ns1:EndDay>2019-02-01</ns1:EndDay>
  <ns1:Time>07:00:00</ns1:Time>
  <ns1:TariffValue>
    <ns1:Index>1</ns1:Index>
    <ns1:High>1</ns1:High>
  </ns1:TariffValue>
</ns1:Schedules>

<ns1:Schedules>
  <ns1:WeekDay>ABSOLUTEDAY</ns1:WeekDay>
  <ns1:StartDay>2018-01-01</ns1:StartDay>
  <ns1:EndDay>2018-01-01</ns1:EndDay>
  <ns1:Time>07:00:00</ns1:Time>
  <ns1:TariffValue>
    <ns1:Index>1</ns1:Index>
    <ns1:High>0</ns1:High>
  </ns1:TariffValue>
</ns1:Schedules>

<ns1:Schedules>
  <ns1:WeekDay>ABSOLUTEDAY</ns1:WeekDay>
  <ns1:StartDay>2018-04-02</ns1:StartDay>
  <ns1:EndDay>2018-04-02</ns1:EndDay>
  <ns1:Time>07:00:00</ns1:Time>
  <ns1:TariffValue>
    <ns1:Index>1</ns1:Index>
    <ns1:High>0</ns1:High>
  </ns1:TariffValue>
</ns1:Schedules>

<ns1:Schedules>
  <ns1:WeekDay>ABSOLUTEDAY</ns1:WeekDay>
  <ns1:StartDay>2018-05-10</ns1:StartDay>
  <ns1:EndDay>2018-05-10</ns1:EndDay>
  <ns1:Time>07:00:00</ns1:Time>
  <ns1:TariffValue>
    <ns1:Index>1</ns1:Index>
    <ns1:High>0</ns1:High>
  </ns1:TariffValue>
</ns1:Schedules>

<ns1:Schedules>
  <ns1:WeekDay>ABSOLUTEDAY</ns1:WeekDay>
  <ns1:StartDay>2018-05-21</ns1:StartDay>
  <ns1:EndDay>2018-05-21</ns1:EndDay>
  <ns1:Time>07:00:00</ns1:Time>
  <ns1:TariffValue>
    <ns1:Index>1</ns1:Index>
    <ns1:High>0</ns1:High>
  </ns1:TariffValue>
</ns1:Schedules>
<ns1:Schedules>
  <ns1:WeekDay ABSOLUTEDAY="ABSOLUTEDAY">
    <ns1:StartDay>2018-12-25</ns1:StartDay>
    <ns1:EndDay>2018-12-25</ns1:EndDay>
    <ns1:Time>07:00:00</ns1:Time>
    <ns1:TariffValue>
      <ns1:Index>1</ns1:Index>
      <ns1:High>0</ns1:High>
    </ns1:TariffValue>
  </ns1:WeekDay>
</ns1:Schedules>

<ns1:Schedules>
  <ns1:WeekDay ABSOLUTEDAY="ABSOLUTEDAY">
    <ns1:StartDay>2018-12-26</ns1:StartDay>
    <ns1:EndDay>2018-12-26</ns1:EndDay>
    <ns1:Time>07:00:00</ns1:Time>
    <ns1:TariffValue>
      <ns1:Index>1</ns1:Index>
      <ns1:High>0</ns1:High>
    </ns1:TariffValue>
  </ns1:WeekDay>
</ns1:Schedules>

<ns1:Schedules>
  <ns1:WeekDay ABSOLUTEDAY="ABSOLUTEDAY">
    <ns1:StartDay>2019-01-01</ns1:StartDay>
    <ns1:EndDay>2019-01-01</ns1:EndDay>
    <ns1:Time>07:00:00</ns1:Time>
    <ns1:TariffValue>
      <ns1:Index>1</ns1:Index>
      <ns1:High>0</ns1:High>
    </ns1:TariffValue>
  </ns1:WeekDay>
</ns1:Schedules>
</ns1:SetScheduleRequest>
</soapenv:Body>
</soapenv:Envelope>

SOAP Response Message:

  <SOAP-ENV:Body>
      <ns2:AsyncResponse>
        <ns3:CorrelationUid>LianderNetManagement||KAI-0000000053||20180926150922041</ns3:CorrelationUid>
        <ns3:DeviceId>KAI-0000000053</ns3:DeviceId>
      </ns2:AsyncResponse>
    </ns2:SetScheduleAsyncResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

SOAP Request message for response:

  <soapenv:Header>
    <ns:ApplicationName>SoapUI</ns:ApplicationName>
    <ns:UserName>Sander</ns:UserName>
    <ns:OrganisationIdentification>Liander NetManagement</ns:OrganisationIdentification>
  </soapenv:Header>
  <ns1:SetScheduleAsyncRequest>
    <ns1:AsyncRequest>
      <ns:CorrelationUid>LianderNetManagement||KAI-0000000053||20180926150922041</ns:CorrelationUid>
      <ns1:DeviceId>KAI-0000000053</ns1:DeviceId>
    </ns1:AsyncRequest>
  </ns1:SetScheduleAsyncRequest>
</soapenv:Body>
IEC61850 protocol adapter logging:

2018-09-26 15:09:22.381 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-26 15:09:22.381 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-26 15:09:22.381 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-26 15:09:22.393 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-26 15:09:22.496 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-26 15:09:22.552 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-26 15:09:22.552 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-26 15:09:22.600 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-26 15:09:22.600 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
GetFirmwareVersion

GetFirmwareVersion messages

Description

Request which queries the device for its current firmware version.

Response which returns the result of the request and, if 'result = OK' contains the firmware version.

IEC61850 Fields

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>FC</th>
<th>SUB ATTRIBUTE</th>
<th>DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWDeviceGenericIO/CSLC.FuncFwDw</td>
<td>ST</td>
<td>curVer</td>
<td>VisString32</td>
<td>Current functional firmware version.</td>
</tr>
<tr>
<td>SWDeviceGenericIO/CSLC.ScyFwDw</td>
<td>ST</td>
<td>curVer</td>
<td>VisString32</td>
<td>Current security firmware version.</td>
</tr>
</tbody>
</table>

Example

Soap requests and responses sent to and from platform:

```xml
<soapenv:Header>
    <com:OrganisationIdentification>LianderNetManagement</com:OrganisationIdentification>
    <com:UserName>Sander</com:UserName>
    <com:ApplicationName>SoapUI</com:ApplicationName>
</soapenv:Header>
<br:message>
    <fman:GetFirmwareVersionRequest xmlns:fman="http://www.opensmartgridplatform.org/schemas/common/firmwaremanagement/2014/10"
        <fman:DeviceIdentification>KAI-0000000053</fman:DeviceIdentification>
    </fman:GetFirmwareVersionRequest>
</soapenv:Body>
</soapenv:Envelope>

<SOAP-ENV:Header/>
<SOAP-ENV:Body>
    <ns2:GetFirmwareVersionAsyncResponse xmlns:ns2="http://www.opensmartgridplatform.org/schemas/common/firmwaremanagement/2014/10"
        <ns2:AsyncResponse>
            <ns3:CorrelationUid>LianderNetManagement|||KAI-0000000053|||20180924135200412</ns3:CorrelationUid>
            <ns3:DeviceId>KAI-0000000053</ns3:DeviceId>
        </ns2:AsyncResponse>
    </ns2:GetFirmwareVersionAsyncResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

<soapenv:Header>
    <com:OrganisationIdentification>LianderNetManagement</com:OrganisationIdentification>
    <com:UserName>Sander</com:UserName>
    <com:ApplicationName>SoapUI</com:ApplicationName>
</soapenv:Header>
<br:message>
        <fman:AsyncRequest>
            <fman:DeviceIdentification>KAI-0000000053</fman:DeviceIdentification>
        </fman:GetFirmwareVersionAsyncRequest>
</soapenv:Body>
</soapenv:Envelope>

<SOAP-ENV:Header/>
<SOAP-ENV:Body>
    <ns2:GetFirmwareVersionResponse xmlns:ns2="http://www.opensmartgridplatform.org/schemas/common/firmwaremanagement/2014/10"
        <ns2:Result>OK</ns2:Result>
        <ns2:FirmwareVersion>Grid eXchange Fabric Documentation</ns2:FirmwareVersion>
</soapenv:Body>
</soapenv:Envelope>
```
Platform message of the data received from the device:

LogicalDevice: SWDeviceGenericIO
messageType: GetFirmwareVersion {
  CSLC.FuncFwDw[ST].curVer: 01_21_01A
  CSLC.ScyFwDw[ST].curVer: 01_06_02A
}

IEC61850 protocol adapter logging:

2018-10-01 14:50:18.281 [dev-box] [iec61850RequestsMessageListenerContainer-2] INFO o.o.a.p.i.i.m.BaseMessageProcessor@printDomainInfo:53 - Calling DeviceService function: GET_FIRMWARE_VERSION for domain: CORE 1.0
2018-10-01 14:50:18.281 [dev-box] [iec61850RequestsMessageListenerContainer-2] INFO o.o.a.p.i.i.m.BaseMessageProcessor@getJmsXdeliveryCount:64 - jmsXdeliveryCount: 1
2018-10-01 14:50:18.281 [dev-box] [iec61850RequestsMessageListenerContainer-2] INFO ... - Trying to connect to deviceIdentification: KAI-0000000053 at IP address 84.30.69.148 using response time-out: 10000
2018-10-01 14:50:18.287 [dev-box] [iec61850RequestsMessageListenerContainer-2] WARN o.o.c.db.api.iec61850.entities.Ssld@createDefaultConfiguration:99 - DeviceType is SSLD, returning default list of DeviceOutputSetting: 1 TARIFF, 2 & 3 & 4 LIGHT
2018-10-01 14:50:18.290 [dev-box] [iec61850RequestsMessageListenerContainer-2] INFO ... - Retrieved internal to external index map for device KAI-0000000053: {0=0, 1=1, 2=2, 3=3, 4=4}
2018-10-01 14:50:18.297 [dev-box] [iec61850RequestsMessageListenerContainer-2] INFO o.o.a.p.i.i.n.Iec61850Client@connect:97 - Attempting to connect to server: 84.30.69.148 on port: 102, max redelivery count: 1 and max retry count: 1
2018-10-01 14:50:18.391 [dev-box] [iec61850RequestsMessageListenerContainer-2] INFO o.o.a.p.i.i.n.Iec61850GetFirmwareVersionCommand@apply:45 - Reading the functional firmware version
2018-10-01 14:50:18.394 [dev-box] [iec61850RequestsMessageListenerContainer-2] INFO o.o.a.p.i.i.n.h.DeviceConnection@createObjectReference:94 - Device: KAI-0000000053, ObjectReference: SWDeviceGenericIO/CSLC.FuncFwDw
2018-10-01 14:50:18.467 [dev-box] [iec61850RequestsMessageListenerContainer-2] INFO o.o.a.p.i.i.n.helper.NodeContainer@getString:88 - Device: KAI-0000000053, curVer has value 01_21_01A
2018-10-01 14:50:18.469 [dev-box] [iec61850RequestsMessageListenerContainer-2] INFO o.o.a.p.i.i.n.s.c.Iec61850GetFirmwareVersionCommand@apply:60 - Reading the security firmware version
2018-10-01 14:50:18.501 [dev-box] [iec61850RequestsMessageListenerContainer-2] INFO o.o.a.p.i.i.n.helper.NodeContainer@getString:88 - Device: KAI-0000000053, curVer has value 01_06_02A
2018-10-01 14:50:18.504 [Thread-67] INFO o.o.a.p.i.i.n.r.Iec61850ClientSSLDEventListener@associationClosed:365 - associationClosed() for device: KAI-0000000053, IOException: Connection disconnected by client
UpdateFirmware

UpdateFirmware messages

Description
Request which commands a device to download and install new firmware. The request contains a URL defining the location of the new firmware image. The device should download the firmware from that location.

Response which returns the result of the request.

IEC61850 Fields

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>FC</th>
<th>SUB ATTRIBUTE</th>
<th>DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSLC.FuncFwDw CF</td>
<td>url</td>
<td></td>
<td>VisString255</td>
<td>Functional firmware. Set new firmware file download URL here, device will download the new firmware file and then replace the old firmware file at startT.</td>
</tr>
<tr>
<td>CSLC.FuncFwDw CF</td>
<td>startT</td>
<td></td>
<td>Timestamp</td>
<td>Functional firmware. Device will try to install new firmware file at this Timestamp (date + time).</td>
</tr>
<tr>
<td>CSLC.ScyFwDw CF</td>
<td>url</td>
<td></td>
<td>VisString255</td>
<td>Security firmware. Set new firmware file download URL here, device will download the new firmware file and then replace the old firmware file at startT.</td>
</tr>
<tr>
<td>CSLC.ScyFwDw CF</td>
<td>startT</td>
<td></td>
<td>Timestamp</td>
<td>Security firmware. Device will try to install new firmware file at this Timestamp (date + time).</td>
</tr>
</tbody>
</table>

Example
Soap requests and responses sent to and from platform:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                 xmlns:ns="http://www.opensmartgridplatform.org/schemas/common/2014/10"
                 xmlns:ns1="http://www.opensmartgridplatform.org/schemas/common/firmwaremanagement/2014"

<soapenv:Header>
    <ns:ApplicationName>SoapUI</ns:ApplicationName>
    <ns:UserName>Sander</ns:UserName>
    <ns:OrganisationIdentification>LianderNetManagement</ns:OrganisationIdentification>
</soapenv:Header>

<soapenv:Body>
    <ns1:UpdateFirmwareRequest>
        <ns1:DeviceIdentification>KAI-0000000053</ns1:DeviceIdentification>
        <ns1:FirmwareIdentification>KAI-SSLD-V2</ns1:FirmwareIdentification>
        <ns1:FirmwareModuleType>FUNCTIONAL</ns1:FirmwareModuleType>
    </ns1:UpdateFirmwareRequest>
</soapenv:Body>
</soapenv:Envelope>


<SOAP-ENV:Header/>

<SOAP-ENV:Body>
    
</SOAP-ENV:Body>

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                 xmlns:ns="http://www.opensmartgridplatform.org/schemas/common/2014/10"
                 xmlns:ns1="http://www.opensmartgridplatform.org/schemas/common/firmwaremanagement/2014/1"

<soapenv:Header>
    <ns:ApplicationName>SoapUI</ns:ApplicationName>
    <ns:UserName>Sander</ns:UserName>
    <ns:OrganisationIdentification>LianderNetManagement</ns:OrganisationIdentification>
</soapenv:Header>
```
<soapenv:Body>
  <ns1:UpdateFirmwareAsyncRequest>
    <ns1:AsyncRequest>
      <ns:DeviceId>KAI-0000000053</ns:DeviceId>
      <ns2:UpdateFirmwareAsyncRequest/>
    </ns1:AsyncRequest>
  </ns1:UpdateFirmwareAsyncRequest>
</soapenv:Body>

Platform message of the data written to the device:
LogicalDevice: SWDeviceGenericIO
messageType: UpdateFirmware {
  CSLC.FuncFwDw[CF].startT: 2018-09-26 10:04:52
  CSLC.FuncFwDw[CF].url: https://168.63.97.65:63443/firmware/KAI-SSLD-V2
}

IEC61850 protocol adapter logging:
SetReboot

SetReboot messages

Description
Request which commands a device to reboot immediately. After the reboot, the device will switch its relays according to its schedule. Any ad hoc changes to relays will be lost.

Response which returns the result of the request.

IEC61850 Fields

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>FC</th>
<th>SUB ATTRIBUTE</th>
<th>DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSLC.RbOper</td>
<td>ST</td>
<td>Oper.ctlVal</td>
<td>BOOLEAN</td>
<td>Flag which, if set to true, will trigger a reboot.</td>
</tr>
</tbody>
</table>

Example

Soap requests and responses sent to and from platform:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:ns="http://www.opensmartgridplatform.org/schemas/common/2014/10"
    xmlns:ns1="http://www.opensmartgridplatform.org/schemas/adhocmanagement/2014/10">
    <soapenv:Header>
        <ns:ApplicationName>SoapUI</ns:ApplicationName>
        <ns:UserName>Sander</ns:UserName>
        <ns:OrganisationIdentification>LianderNetManagement</ns:OrganisationIdentification>
    </soapenv:Header>
    <soapenv:Body>
        <ns1:SetRebootRequest>
            <ns1:DeviceIdentification>KAI-0000000053</ns1:DeviceIdentification>
        </ns1:SetRebootRequest>
    </soapenv:Body>
</soapenv:Envelope>
```

```xml
    xmlns:ns2="http://www.opensmartgridplatform.org/schemas/adhocmanagement/2014/10"
    xmlns:ns3="http://www.opensmartgridplatform.org/schemas/common/2014/10">
    <ns2:SetRebootAsyncResponse>
        <ns3:CorrelationUid>LianderNetManagement|||KAI-0000000053|||20180925104202472</ns3:CorrelationUid>
        <ns3:DeviceId>KAI-0000000053</ns3:DeviceId>
    </ns2:SetRebootAsyncResponse>
</SOAP-ENV:Envelope>
```

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:ns="http://www.opensmartgridplatform.org/schemas/common/2014/10"
    xmlns:ns1="http://www.opensmartgridplatform.org/schemas/adhocmanagement/2014/10">
    <soapenv:Header>
        <ns:ApplicationName>SoapUI</ns:ApplicationName>
        <ns:UserName>Sander</ns:UserName>
        <ns:OrganisationIdentification>LianderNetManagement</ns:OrganisationIdentification>
    </soapenv:Header>
    <soapenv:Body>
        <ns1:SetRebootAsyncRequest>
            <ns1:AsyncRequest>
                <ns:CorrelationUid>LianderNetManagement|||KAI-0000000053|||20180925104202472</ns:CorrelationUid>
                <ns:DeviceId>KAI-0000000053</ns:DeviceId>
            </ns1:AsyncRequest>
        </ns1:SetRebootAsyncRequest>
    </soapenv:Body>
</soapenv:Envelope>
```
<SOAP-ENV:Body>
  <ns2:SetRebootResponse
    xmlns:ns2="http://www.opensmartgridplatform.org/schemas/adhocmanagement/2014/10"
    xmlns:ns3="http://www.opensmartgridplatform.org/schemas/common/2014/10">
    <ns2:Result>OK</ns2:Result>
  </ns2:SetRebootResponse>
</SOAP-ENV:Body>

IEC61850 platform message of the data set on the device:

LogicalDevice: SWDeviceGenericIO
messageType: Reboot {
  CSLC.RbOper[ST].Oper.ctlVal: true
}

IEC61850 protocol adapter logging:

2018-09-25 10:44:06.142] [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
SwDeviceGenericIO/CSLC.RbOper.Oper [CO]
SwDeviceGenericIO/CSLC.RbOper.Oper.ctlVal: false
SwDeviceGenericIO/CSLC.RbOper.Oper.origin [CO]
SwDeviceGenericIO/CSLC.RbOper.Oper.origin.orCat: 0
SwDeviceGenericIO/CSLC.RbOper.Oper.origin.orIdent: []
SwDeviceGenericIO/CSLC.RbOper.Oper.ctlNum: 0
SwDeviceGenericIO/CSLC.RbOper.Oper.T: Thu Jan 01 00:00:00 UTC 1970
SwDeviceGenericIO/CSLC.RbOper.Oper.Test: false
SwDeviceGenericIO/CSLC.RbOper.Oper.Check: 0x0
2018-09-25 10:44:06.160] [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
SwDeviceGenericIO/CSLC.RbOper.Oper.ctlVal: false
SwDeviceGenericIO/CSLC.RbOper.Oper.origin [CO]
SwDeviceGenericIO/CSLC.RbOper.Oper.origin.orCat: 0
SwDeviceGenericIO/CSLC.RbOper.Oper.origin.orIdent: []
SwDeviceGenericIO/CSLC.RbOper.Oper.ctlNum: 0
SwDeviceGenericIO/CSLC.RbOper.Oper.T: Thu Jan 01 00:00:00 UTC 1970
SwDeviceGenericIO/CSLC.RbOper.Oper.Test: false
SwDeviceGenericIO/CSLC.RbOper.Oper.Check: 0x0
2018-09-25 10:44:06.176] [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
SwDeviceGenericIO/CSLC.RbOper.Oper.ctlVal: false
SwDeviceGenericIO/CSLC.RbOper.Oper.origin [CO]
SwDeviceGenericIO/CSLC.RbOper.Oper.origin.orCat: 0
SwDeviceGenericIO/CSLC.RbOper.Oper.origin.orIdent: []
SwDeviceGenericIO/CSLC.RbOper.Oper.ctlNum: 0
SwDeviceGenericIO/CSLC.RbOper.Oper.T: Thu Jan 01 00:00:00 UTC 1970
SwDeviceGenericIO/CSLC.RbOper.Oper.Test: false
SwDeviceGenericIO/CSLC.Rb Oper.Oper.Check: 0x0
2018-09-25 10:44:06.176] [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
SwDeviceGenericIO/CSLC.RbOper.Oper.ctlVal: false
SwDeviceGenericIO/CSLC.RbOper.Oper.origin [CO]
SwDeviceGenericIO/CSLC.RbOper.Oper.origin.orCat: 0
SwDeviceGenericIO/CSLC.RbOper.Oper.origin.orIdent: []
SwDeviceGenericIO/CSLC.RbOper.Oper.ctlNum: 0
SwDeviceGenericIO/CSLC.RbOper.Oper.T: Thu Jan 01 00:00:00 UTC 1970
SwDeviceGenericIO/CSLC.RbOper.Oper.Test: false
SwDeviceGenericIO/CSLC.RbOper.Oper.Check: 0x0
2018-09-25 10:44:06.176] [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
StartSelfTest

StartSelfTest messages

Description
Request which commands the device to switch all light relays on and then queries the device for the status of the relays.
OGSP checks if the status of the relays is as expected (on in this case).
Response returns the result of the request.

IEC61850 Fields

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>FC</th>
<th>SUB ATTRIBUTE</th>
<th>DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>XSWC2.Pos</td>
<td>CO</td>
<td>Oper.ctlVal</td>
<td>BOOLEAN</td>
<td>Flag which, if set to true, immediately switches relay 2 on.</td>
</tr>
<tr>
<td>XSWC3.Pos</td>
<td>CO</td>
<td>Oper.ctlVal</td>
<td>BOOLEAN</td>
<td>Flag which, if set to true, immediately switches relay 3 on.</td>
</tr>
</tbody>
</table>

Example
Soap requests and responses sent to and from platform:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:ns="http://www.opensmartgridplatform.org/schemas/common/2014/10"
 xmlns:ns1="http://www.opensmartgridplatform.org/schemas/deviceinstallation/2014/10">
  <soapenv:Header>
    <ns:ApplicationName>SoapUI</ns:ApplicationName>
    <ns:UserName>Sander</ns:UserName>
    <ns:OrganisationIdentification>LianderNetManagement</ns:OrganisationIdentification>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:StartDeviceTestRequest>
      <ns1:DeviceIdentification>KAI-0000000053</ns1:DeviceIdentification>
    </ns1:StartDeviceTestRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:ns="http://www.opensmartgridplatform.org/schemas/common/2014/10"
 xmlns:ns1="http://www.opensmartgridplatform.org/schemas/deviceinstallation/2014/10">
  <soapenv:Header>
    <ns:ApplicationName>SoapUI</ns:ApplicationName>
    <ns:UserName>Sander</ns:UserName>
    <ns:OrganisationIdentification>LianderNetManagement</ns:OrganisationIdentification>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:StartDeviceTestAsyncRequest>
      <ns1:AsyncRequest>
        <ns:CorrelationUid>LianderNetManagement|||KAI-0000000053|||20180925142331454</ns:CorrelationUid>
        <ns:DeviceId>KAI-0000000053</ns:DeviceId>
      </ns1:AsyncRequest>
    </ns1:StartDeviceTestAsyncRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:ns="http://www.opensmartgridplatform.org/schemas/common/2014/10"
 xmlns:ns1="http://www.opensmartgridplatform.org/schemas/deviceinstallation/2014/10">
  <soapenv:Header>
    <ns:ApplicationName>SoapUI</ns:ApplicationName>
    <ns:UserName>Sander</ns:UserName>
    <ns:OrganisationIdentification>LianderNetManagement</ns:OrganisationIdentification>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:StartDeviceTestAsyncRequest>
      <ns1:AsyncRequest>
        <ns:CorrelationUid>LianderNetManagement|||KAI-0000000053|||20180925142331454</ns:CorrelationUid>
        <ns:DeviceId>KAI-0000000053</ns:DeviceId>
      </ns1:AsyncRequest>
    </ns1:StartDeviceTestAsyncRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

```
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:StartDeviceTestAsyncResponse>
      <ns3:AsyncResponse>
        <ns3:CorrelationUid>LianderNetManagement|||KAI-0000000053|||20180925142331454</ns3:CorrelationUid>
        <ns3:DeviceId>KAI-0000000053</ns3:DeviceId>
      </ns3:AsyncResponse>
    </ns2:StartDeviceTestAsyncResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
IEC61850 platform message of the data written to the device:

LogicalDevice: SWDeviceGenericIO
messageType: StartSelfTest
  XSWC2.Pos[CO].Oper.ctlVal: true
  XSWC3.Pos[CO].Oper.ctlVal: true

IEC61850 platform message of the data read from the device:

LogicalDevice: SWDeviceGenericIO
messageType: GetStatus
  XSWC2.Pos[ST].stVal: true
  CSLC.SWCF[CF].LT: RELAY
  XSWC3.Pos[ST].stVal: true
  CSLC.SWC[CF].LT: RELAY

IEC61850 protocol adapter logging:

2018-09-25 14:23:32.005 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o.o.a.p.i.i.n.Iec61850Client@connect:97 - Attempting to connect to server: 84.30.69.148 on port: 102, max redelivery count: 3 and max retry count: 3
2018-09-25 14:23:32.002 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o.o.a.p.i.i.n.Iec61850Client@connect:113 - Connected to device: KAI-0000000053
2018-09-25 14:23:31.995 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o.o.a.p.i.i.n.Iec61850Client@connect:113 - Turns all lights relays on
2018-09-25 14:23:31.994 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o.o.a.p.i.i.m.DeviceRequestMessageListener@onMessage:61 - Received message of type: START_SELF_TEST with message priority: 4
2018-09-25 14:23:32.056 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o.o.a.p.i.i.n.Iec61850Client@connect:113 - Connected to device: KAI-0000000053
2018-09-25 14:23:32.005 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o.o.a.p.i.i.n.Iec61850Client@connect:113 - Connected to device: KAI-0000000053
2018-09-25 14:23:31.995 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o.o.a.p.i.i.m.DeviceRequestMessageListener@onMessage:61 - Received message of type: START_SELF_TEST with message priority: 4
2018-09-25 14:23:32.523 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o.o.a.p.i.i.m.DeviceRequestMessageListener@onMessage:61 - Received message of type: START_SELF_TEST with message priority: 4
2018-09-25 14:23:32.162 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o.o.a.p.i.i.m.DeviceRequestMessageListener@onMessage:61 - Received message of type: START_SELF_TEST with message priority: 4
2018-09-25 14:23:32.128 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o.o.a.p.i.i.m.DeviceRequestMessageListener@onMessage:61 - Received message of type: START_SELF_TEST with message priority: 4
2018-09-25 14:23:32.056 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o.o.a.p.i.i.m.DeviceRequestMessageListener@onMessage:61 - Received message of type: START_SELF_TEST with message priority: 4

<SOAP-ENV:Body>
  <ns2:StartDeviceTestResponse
    xmlns:ns2="http://www.opensmartgridplatform.org/schemas/deviceinstallation/2014/10"
    xmlns:ns3="http://www.opensmartgridplatform.org/schemas/common/2014/10">
    <ns2:Result>OK</ns2:Result>
  </ns2:StartDeviceTestResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
StopSelfTest

StopSelfTest messages

Description
Request which commands the device to switch all light relays off and then queries the device for the status of the relays. OGSP checks if the status of the relays is as expected (off in this case).

Response which returns the result of the request.

IEC61850 Fields

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>FC</th>
<th>SUB ATTRIBUTE</th>
<th>DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>XSWC2.Pos</td>
<td>CO</td>
<td>Oper.ctlVal</td>
<td>BOOLEAN</td>
<td>Flag which, if set to false, immediately switches relay 2 off.</td>
</tr>
<tr>
<td>XSWC3.Pos</td>
<td>CO</td>
<td>Oper.ctlVal</td>
<td>BOOLEAN</td>
<td>Flag which, if set to false, immediately switches relay 3 off.</td>
</tr>
</tbody>
</table>

Example

Soap requests and responses sent to and from platform:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
 xmlns:ns="http://www.opensmartgridplatform.org/schemas/common/2014/10"
 xmlns:ns1="http://www.opensmartgridplatform.org/schemas/deviceinstallation/2014/10">
  <soapenv:Header>
    <ns:ApplicationName>SoapUI</ns:ApplicationName>
    <ns:UserName>Sander</ns:UserName>
    <ns:OrganisationIdentification>LianderNetManagement</ns:OrganisationIdentification>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:StopDeviceTestRequest>
      <ns1:DeviceIdentification>KAI-0000000053</ns1:DeviceIdentification>
    </ns1:StopDeviceTestRequest>
  </soapenv:Body>
</soapenv:Envelope>

 xmlns:ns="http://www.opensmartgridplatform.org/schemas/common/2014/10"
 xmlns:ns1="http://www.opensmartgridplatform.org/schemas/deviceinstallation/2014/10">
  <SOAP-ENV:Header>
    <ns:ApplicationName>SoapUI</ns:ApplicationName>
    <ns:UserName>Sander</ns:UserName>
    <ns:OrganisationIdentification>LianderNetManagement</ns:OrganisationIdentification>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <ns1:StopDeviceTestAsyncRequest>
      <ns1:AsyncRequest>
        <ns2:CorrelationUid>LianderNetManagement||KAI-0000000053||20180925142825021</ns2:CorrelationUid>
        <ns1:DeviceId>KAI-0000000053</ns1:DeviceId>
      </ns1:AsyncRequest>
    </ns1:StopDeviceTestAsyncRequest>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
 xmlns:ns="http://www.opensmartgridplatform.org/schemas/common/2014/10"
 xmlns:ns1="http://www.opensmartgridplatform.org/schemas/deviceinstallation/2014/10">
  <soapenv:Header>
    <ns:ApplicationName>SoapUI</ns:ApplicationName>
    <ns:UserName>Sander</ns:UserName>
    <ns:OrganisationIdentification>LianderNetManagement</ns:OrganisationIdentification>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:StopDeviceTestAsyncRequest>
      <ns1:AsyncRequest>
        <ns1:DeviceIdentification>KAI-0000000053</ns1:DeviceIdentification>
      </ns1:AsyncRequest>
    </ns1:StopDeviceTestAsyncRequest>
  </soapenv:Body>
</soapenv:Envelope>

 xmlns:ns="http://www.opensmartgridplatform.org/schemas/common/2014/10"
 xmlns:ns1="http://www.opensmartgridplatform.org/schemas/deviceinstallation/2014/10">
  <SOAP-ENV:Header/>
</SOAP-ENV:Envelope>
```
LogicalDevice: SWDeviceGenericIO
messageType: StopSelfTest {
    XSWC2.Pos[CO].Oper.ctlVal: false
    XSWC3.Pos[CO].Oper.ctlVal: false
}

LogicalDevice: SWDeviceGenericIO
messageType: GetStatus {
    XSWC2.Pos[ST].stVal: false
    CSLC.SWCf[CF].LT: RELAY
    XSWC3.Pos[ST].stVal: false
    XSWC1.Pos[ST].stVal: false
    CSLC.EvnBuf[CF].enbEvnType: 1FFFFFF
}

IEC61850 platform message of the data written to the device:

LogicalDevice: SWDeviceGenericIO
messageType: StopSelfTest {
    XSWC2.Pos[CO].Oper.ctlVal: false
    XSWC3.Pos[CO].Oper.ctlVal: false
}

IEC61850 platform message of the data read from the device:

LogicalDevice: SWDeviceGenericIO
messageType: GetStatus {
    XSWC2.Pos[ST].stVal: false
    CSLC.SWCf[CF].LT: RELAY
    XSWC3.Pos[ST].stVal: false
    XSWC1.Pos[ST].stVal: false
    CSLC.EvnBuf[CF].enbEvnType: 1FFFFFF
}
2018-09-25 14:30:38.486  [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-25 14:30:38.486  [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-25 14:30:38.486  [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-25 14:30:38.486  [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-25 14:30:38.486  [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-25 14:30:38.486  [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-25 14:30:38.486  [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
SetLight

SetLight messages

Description

Request which commands the device to switch on or off one or several light relays. If optional value 'index' is omitted, all relays configured as light are switched. In that case, the message will contain exactly one LightValue and all relays will switch to that LightValue. In case the value 'index' is included, multiple instances of LightValue can be used (up to 6), each indicating a particular relay.

Response which returns the result of the request.

IEC61850 Fields

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>FC</th>
<th>SUB ATTRIBUTE</th>
<th>DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>XSWC2.Pos</td>
<td>CO</td>
<td>Oper.ctlVal</td>
<td>BOOLEAN</td>
<td>Flag which, if set to true, immediately switches relay 2 on.</td>
</tr>
<tr>
<td>XSWC3.Pos</td>
<td>CO</td>
<td>Oper.ctlVal</td>
<td>BOOLEAN</td>
<td>Flag which, if set to true, immediately switches relay 3 on.</td>
</tr>
<tr>
<td>XSWC4.Pos</td>
<td>CO</td>
<td>Oper.ctlVal</td>
<td>BOOLEAN</td>
<td>Flag which, if set to true, immediately switches relay 4 on.</td>
</tr>
</tbody>
</table>

Example

Soap requests and responses sent to and from platform:

```xml
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns="http://www.opensmartgridplatform.org/schemas/common/2014/10"
  xmlns:ns1="http://www.opensmartgridplatform.org/schemas/publiclighting/adhocmanagement/2"
>
  <soapenv:Header>
    <ns:ApplicationName>SoapUI</ns:ApplicationName>
    <ns:UserName>Sander</ns:UserName>
    <ns:OrganisationIdentification>LianderNetManagement</ns:OrganisationIdentification>
  </soapenv:Header>

  <soapenv:Body>
    <ns1:SetLightRequest>
      <ns1:DeviceIdentification>KAI-0000000053</ns1:DeviceIdentification>
      <ns1:LightValue>
        <ns1:On>true</ns1:On>
      </ns1:LightValue>
    </ns1:SetLightRequest>
  </soapenv:Body>
</soapenv:Envelope>

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header />
  <SOAP-ENV:Body>
    <ns2:SetLightAsyncResponse
      xmlns:ns2="http://www.opensmartgridplatform.org/schemas/publiclighting/adhocmanagement/2"
      xmlns:ns3="http://www.opensmartgridplatform.org/schemas/common/2014/10">
      <ns2:AsyncResponse>
        <ns3:CorrelationUid>LianderNetManagement||KAI-0000000053||20180925093054301</ns3:CorrelationUid>
        <ns3:DeviceId>KAI-0000000053</ns3:DeviceId>
      </ns2:AsyncResponse>
    </ns2:SetLightAsyncResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns="http://www.opensmartgridplatform.org/schemas/common/2014/10"
  xmlns:ns1="http://www.opensmartgridplatform.org/schemas/publiclighting/adhocmanagement/2"
>
  <soapenv:Header>
    <ns:ApplicationName>SoapUI</ns:ApplicationName>
    <ns:UserName>Sander</ns:UserName>
    <ns:OrganisationIdentification>LianderNetManagement</ns:OrganisationIdentification>
  </soapenv:Header>

  <soapenv:Body>
    <ns1:SetLightAsyncRequest>
      ...
    </ns1:SetLightAsyncRequest>
  </soapenv:Body>
</soapenv:Envelope>
```
<ns1:AsyncRequest>
  <ns:CorrelationUid>LianderNetManagement|||KAI-0000000053|||20180925093054301</ns:CorrelationUid>
  <ns:DeviceId>KAI-0000000053</ns:DeviceId>
</ns1:AsyncRequest>
</soapenv:Body>
</soapenv:Envelope>

  xmlns:ns2="http://www.opensmartgridplatform.org/schemas/publiclighting/adhocmanager"
  xmlns:ns3="http://www.opensmartgridplatform.org/schemas/common/2014/10">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:SetLightResponse xmlns:ns2="http://www.opensmartgridplatform.org/schemas/publiclighting/adhocmanager"
      xmlns:ns3="http://www.opensmartgridplatform.org/schemas/common/2014/10">
      <ns2:Result>OK</ns2:Result>
    </ns2:SetLightResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

IEC61850 platform message of the data set on the device:

LogicalDevice: SWDeviceGenericIO
messageType: SetLight {
  XSWC2.Pos[CO].Oper ctlVal: true
  XSWC3.Pos[CO].Oper ctlVal: true
}

IEC61850 protocol adapter logging:

2018-09-25 09:30:54.775] [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-25 09:30:54.782] [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-25 09:30:54.783] [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-25 09:30:54.799] [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-25 09:30:54.852] [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-25 09:30:54.881] [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-25 09:30:54.898] [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-25 09:30:54.986] [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-25 09:30:54.986] [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-25 09:30:55.211] [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-25 09:30:55.553] [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
2018-09-25 09:30:55.554] [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o
SetTransition

SetTransition messages

Description

Request which informs a device of a daylight transition: it has become dark (sunset) or light (sunrise). The device will switch the relays, which have schedule entries for transition messages. See light schedule-entry for more information regarding switch schedules.

Response which returns the result of the request.

IEC61850 Fields

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>SUB ATTRIBUTE</th>
<th>DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSLC.Sensor</td>
<td>CO Oper.ctlVal</td>
<td>BOOLEAN</td>
<td>Flag indicating transition type, true = DAY_NIGHT, false = NIGHT_DAY.</td>
</tr>
</tbody>
</table>

Example

Soap requests and responses sent to and from platform:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns="http://www.opensmartgridplatform.org/schemas/common/2014/10"
  xmlns:ns1="http://www.opensmartgridplatform.org/schemas/publiclighting/adhocmanagement/2014/10"
><soapenv:Header>
  <ns:ApplicationName>SoapUI</ns:ApplicationName>
  <ns:UserName>Sander</ns:UserName>
  <ns:OrganisationIdentification>LianderNetManagement</ns:OrganisationIdentification>
</soapenv:Header>
<soapenv:Body>
  <ns1:SetTransitionRequest>
    <ns1:DeviceIdentification>KAI-0000000053</ns1:DeviceIdentification>
    <ns1:TransitionType>NIGHT_DAY</ns1:TransitionType>
    <ns1:Time>08:00:00</ns1:Time>
  </ns1:SetTransitionRequest>
</soapenv:Body>
</soapenv:Envelope>

><SOAP-ENV:Header>
  <ns3:CorrelationUid>LianderNetManagement||KAI-0000000053||20180926091217778</ns3:CorrelationUid>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
  <ns2:SetTransitionAsyncResponse>
    <ns3:DeviceId>KAI-0000000053</ns3:DeviceId>
  </ns2:SetTransitionAsyncResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns="http://www.opensmartgridplatform.org/schemas/common/2014/10"
><soapenv:Header>
  <ns:ApplicationName>SoapUI</ns:ApplicationName>
  <ns:UserName>Sander</ns:UserName>
  <ns:OrganisationIdentification>LianderNetManagement</ns:OrganisationIdentification>
</soapenv:Header>
<soapenv:Body>
  <ns1:SetTransitionAsyncRequest>
    <ns1:DeviceIdentification>KAI-0000000053</ns1:DeviceIdentification>
    <ns1:Time>08:00:00</ns1:Time>
  </ns1:SetTransitionAsyncRequest>
</soapenv:Body>
</soapenv:Envelope>

><SOAP-ENV:Header>
  <ns:ApplicationName>SoapUI</ns:ApplicationName>
  <ns:UserName>Sander</ns:UserName>
  <ns:OrganisationIdentification>LianderNetManagement</ns:OrganisationIdentification>
</soapenv:Header>
<soapenv:Body>
  <ns1:SetTransitionAsyncRequest>
    <ns1:DeviceIdentification>KAI-0000000053</ns1:DeviceIdentification>
    <ns1:Time>08:00:00</ns1:Time>
  </ns1:SetTransitionAsyncRequest>
</soapenv:Body>
</SOAP-ENV:Envelope>
```
Platform message of the data written to the device:

LogicalDevice: SWDeviceGenericIO

messageType: SetTransition {
  CSLC.Sensor[CO].Oper.ctlVal: false
}

IEC61850 protocol adapter logging:

2018-09-26 09:12:18.167 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o.o.o.a.p.i.i.n.Iec61850Client@connect:113 - Connected to device: KAI-0000000053

2018-09-26 09:12:18.121 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o.o.o.a.p.i.i.n.Iec61850Client@connect:97 - Attempting to connect to server: 84.30.69.148 on port: 102, max redelivery count: 3 and max retry count: 3

2018-09-26 09:12:18.116 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO ... - Trying to connect to deviceIdentification: KAI-0000000053 at IP address 84.30.69.148 using response time-out: 10000

2018-09-26 09:12:18.111 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o.o.a.p.i.i.m.BaseMessageProcessor@getJmsXdeliveryCount:64 - jmsXdeliveryCount: 1

2018-09-26 09:12:18.111 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o.o.a.p.i.i.m.BaseMessageProcessor@printDomainInfo:53 - Calling DeviceService function: SET_TRANSITION for domain: PUBLIC_LIGHTING 1.0

2018-09-26 09:12:18.111 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-11] INFO o.o.a.p.i.i.m.DeviceRequestMessageListener@onMessage:61 - Received message of type: SET_TRANSITION with message priority: 4

SOAP-ENV:Envelope
  <ns2:Result>
    <ns2:Details>
      <ns2:NOT FOUND/>
    </ns2:Details>
  </ns2:Result>
</ns2:SetTransitionResponse>
</SOAP-ENV:Envelope>

Platform message of the data written to the device:

LogicalDevice: SWDeviceGenericIO

messageType: SetTransition {
  CSLC.Sensor[CO].Oper.ctlVal: false
}
GetStatus

GetStatus messages

Description
Request which queries the device for the status of all relays, the type of configuration, and the event notification mask set on the device.

Response which returns the result of the request and, if 'result = OK', contains the current status for all of the relays and other information.

IEC61850 Fields

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>FC</th>
<th>Sub Attribute</th>
<th>DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>XSWC1.Pos</td>
<td>ST</td>
<td>stVal</td>
<td>BOOLEAN</td>
<td>Current switch status for relay 1.</td>
</tr>
<tr>
<td>XSWC2.Pos</td>
<td>ST</td>
<td>stVal</td>
<td>BOOLEAN</td>
<td>Current switch status for relay 2.</td>
</tr>
<tr>
<td>XSWC3.Pos</td>
<td>ST</td>
<td>stVal</td>
<td>BOOLEAN</td>
<td>Current switch status for relay 3.</td>
</tr>
<tr>
<td>CSLC.EvnBuf</td>
<td>CF</td>
<td>enbEvnType</td>
<td>VisString32</td>
<td>Bitmask indicating which event notification types are enabled.</td>
</tr>
<tr>
<td>CSLC.SWCf</td>
<td>CF</td>
<td>LT</td>
<td>VisString64</td>
<td>Device light type, always &quot;RELAY&quot;.</td>
</tr>
</tbody>
</table>

Example
Soap requests and responses sent to and from platform:

```xml
  <SOAP-ENV:Header>
  <OrganisationIdentification xmlns="http://www.opensmartgridplatform.org/schemas/common"/>
  <ApplicationName xmlns="http://www.opensmartgridplatform.org/schemas/common">SoapUI</ApplicationName>
  <UserName xmlns="http://www.opensmartgridplatform.org/schemas/common">Sander</UserName>
  </SOAP-ENV:Header>
  <ns2:GetStatusRequest xmlns:ns2="http://www.opensmartgridplatform.org/schemas/publiclighting/adhocmanagement/2014/10">
    <ns2:DeviceIdentification>KAI-0000000053</ns2:DeviceIdentification>
  </ns2:GetStatusRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

  <SOAP-ENV:Header>
  <OrganisationIdentification xmlns="http://www.opensmartgridplatform.org/schemas/common"/>
  <ApplicationName xmlns="http://www.opensmartgridplatform.org/schemas/common">SoapUI</ApplicationName>
  <UserName xmlns="http://www.opensmartgridplatform.org/schemas/common">Sander</UserName>
  </SOAP-ENV:Header>
  <ns2:GetStatusAsyncRequest xmlns:ns2="http://www.opensmartgridplatform.org/schemas/publiclighting/adhocmanagement/2014/10">
    <ns2:AsyncRequest>
      <ns3:CorrelationUid>LianderNetManagement|||KAI-0000000053|||20180924111517726</ns3:CorrelationUid>
      <ns3:DeviceId>KAI-0000000053</ns3:DeviceId>
    </ns2:AsyncRequest>
  </ns2:GetStatusAsyncRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

  <SOAP-ENV:Header>
  <OrganisationIdentification xmlns="http://www.opensmartgridplatform.org/schemas/common"/>
  <ApplicationName xmlns="http://www.opensmartgridplatform.org/schemas/common">SoapUI</ApplicationName>
  <UserName xmlns="http://www.opensmartgridplatform.org/schemas/common">Sander</UserName>
  </SOAP-ENV:Header>
  <ns2:GetStatusAsyncRequest xmlns:ns2="http://www.opensmartgridplatform.org/schemas/publiclighting/adhocmanagement/2014/10">
    <ns2:AsyncRequest>
      <ns3:CorrelationUid>LianderNetManagement|||KAI-0000000053|||20180924111517726</ns3:CorrelationUid>
      <ns3:DeviceId>KAI-0000000053</ns3:DeviceId>
    </ns2:AsyncRequest>
  </ns2:GetStatusAsyncRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

  <SOAP-ENV:Header>
  <OrganisationIdentification xmlns="http://www.opensmartgridplatform.org/schemas/common"/>
  <ApplicationName xmlns="http://www.opensmartgridplatform.org/schemas/common">SoapUI</ApplicationName>
  <UserName xmlns="http://www.opensmartgridplatform.org/schemas/common">Sander</UserName>
  </SOAP-ENV:Header>
  <ns2:GetStatusAsyncResponse xmlns:ns2="http://www.opensmartgridplatform.org/schemas/publiclighting/adhocmanagement/2014/10">
    <ns2:AsyncResponse>
      <ns3:CorrelationUid>LianderNetManagement|||KAI-0000000053|||20180924111517726</ns3:CorrelationUid>
      <ns3:DeviceId>KAI-0000000053</ns3:DeviceId>
    </ns2:AsyncResponse>
  </ns2:GetStatusAsyncResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
Platform message of the data read from the device:

LogicalDevice: SWDeviceGenericIO
messageType: GetStatus

{ XSWC2.Pos[ST].stVal: false
CSLC.WSC[CF].LT: RELAY
XSWC3.Pos[ST].stVal: false
XSWC1.Pos[ST].stVal: false
CSLC.EvnBuf[CF].enbEvnType: 1FFFFFF
}

IEC61850 protocol Adapter logging:

2018-10-01 13:31:42.182 [dev-box] [iecb61850RequestsMessageListenerContainer-2] INFO o.o.a.p.i.i.m.SsldDeviceRequestMessageProcessor@handleGetStatusDeviceResponse:66 - Handling getStatusDeviceResponse for device: KAI-0000000053
2018-10-01 13:31:42.182 [dev-box] [iecb61850RequestsMessageListenerContainer-2] INFO o.o.a.p.i.i.n.helper.NodeContainer@getString:88 - Device: KAI-0000000053, LT has value RELAY
UpdateDeviceSslCertification

UpdateDeviceSslCertification messages

Description
Request to download a new SSL certificate from the certificate server. The device will be given the domain name and URL where the certificate is located.

IEC61850 Fields

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>SUB ATTRIBUTE</th>
<th>DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSLC.CARepl</td>
<td>CF url</td>
<td>VisString255</td>
<td>Set new CA file download URL here, device will download the new CA file and then replace the old CA file at startT.</td>
</tr>
<tr>
<td>CSLC.CARepl</td>
<td>CF startT</td>
<td>VisString32</td>
<td>Device will download CA file at this Timestamp (date + time).</td>
</tr>
</tbody>
</table>

Example
Soap requests and responses sent to and from platform:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="http://www.opensmartgridplatform.org/schemas/common/2014/10"
xmlns:ns1="http://www.opensmartgridplatform.org/schemas/devicemanagement/2014/10">
  <soapenv:Header>
    <ns:ApplicationName>SoapUI</ns:ApplicationName>
    <ns:UserName>Kevin</ns:UserName>
    <ns:OrganisationIdentification>LianderNetManagement</ns:OrganisationIdentification>
  </soapenv:Header>
  <ns1:UpdateDeviceSslCertificationRequest>
    <ns1:DeviceIdentification>KAI-0000000053</ns1:DeviceIdentification>
    <ns1:Certificate>
      <ns1:certificateDomain>cert-server</ns1:certificateDomain>
      <ns1:certificateUrl>/certs/new-cert.pem</ns1:certificateUrl>
    </ns1:Certificate>
  </ns1:UpdateDeviceSslCertificationRequest>
</soapenv:Envelope>
```

```
xmlns:ns1="http://www.opensmartgridplatform.org/schemas/devicemanagement/2014/10">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:UpdateDeviceSslCertificationAsyncResponse xmlns:ns2="http://www.opensmartgridplatform.org/schemas/devicemanagement/2014/10">
      <ns3:CorrelationUid>LianderNetManagement|||KAI-0000000053|||20181001092825280</ns3:CorrelationUid>
      <ns3:DeviceId>KAI-0000000053</ns3:DeviceId>
    </ns2:UpdateDeviceSslCertificationAsyncResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="http://www.opensmartgridplatform.org/schemas/common/2014/10"
xmlns:ns1="http://www.opensmartgridplatform.org/schemas/devicemanagement/2014/10">
  <soapenv:Header>
    <ns:ApplicationName>SoapUI</ns:ApplicationName>
    <ns:UserName>Kevin</ns:UserName>
    <ns:OrganisationIdentification>LianderNetManagement</ns:OrganisationIdentification>
  </soapenv:Header>
  <ns1:UpdateDeviceSslCertificationAsyncRequest>
    <ns:CorrelationUid>LianderNetManagement|||KAI-0000000053|||20181001092825280</ns:CorrelationUid>
    <ns:DeviceId>KAI-0000000053</ns:DeviceId>
  </ns1:UpdateDeviceSslCertificationAsyncRequest>
</soapenv:Envelope>
```
IEC61850 platform message of the data sent to the device:

LogicalDevice: SWDeviceGenericIO
messageType: UpdateSslCertificate {
        CSLC.CARepl[CF].url: cert-server/certs/new-cert.pem
        CSLC.CARepl[CF].startT: 2018-10-01 11:36:08
}

IEC61850 protocol Adapter logging:

```
2018-10-01 09:28:25.800] [osgp-tst-03] [iec61850RequestsMessageListenerContainer-13] INFO o.o.a.p.i.i.n.s.c.Iec61850UpdateSslCertificateCommand@apply:78 - Updating the certificate download start time to: Mon Oct 01 11:36:08 UTC 2018
2018-10-01 09:28:25.968] [osgp-tst-03] [iec61850RequestsMessageListenerContainer-13] INFO o.o.a.p.i.i.n.s.c.Iec61850UpdateSslCertificateCommand@apply:44 - Reading the certificate authority url
```

Grid eXchange Fabric Documentation
FlexOVL_540_171101_2

Contract

Contract for FlexOVL_540_171101_2 The contract specifies the messages which can be exchanged with a LMD.

Messages

The messages below are part of OSGP and implemented in the IEC61850 protocol adapter and supported by the FlexOVL_540_171101_2 device firmware.

- **GetStatusRequest** (from platform to device) is a request which queries a device for the current status of all its digital inputs.
- **GetStatusResponse** (from device to platform) is a response which returns the result of the GetStatusRequest and, if result = OK, returns the current status for all of the digital inputs.
- **EventNotificationRequest** (from device to platform) is a request that pushes an event notification from a device to the platform.

No other message types are supported by this device.
FlexOVL_540_171101_2_out.icd

SSLD ICD file, FlexOVL_540_171101_2_out.icd

<?xml version="1.0" encoding="utf-8"?>

<?xml version="1.0" encoding="utf-8"?>

xmlns="http://www.iec.ch/61850/2003/Sldcore"
xmlns:scl="http://www.iec.ch/61850/2003/Scl"
xmlns:esld="http://www.iec.ch/61850/2003/FlexOVL"
xmlns:sxy="http://www.iec.ch/61850/2003/FlexOVL"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

<Private type="ABB SLD">
<esld:Sld>
<esld:Sizes elementSize="1" />
<esld:AlarmSettings />
<esld:MeasurementStatus />
<esld:DefaultColors />
<esld:NameDisplay />
<esld:Fonts />
<esld:MeasurementPrecision defaultDecimals="0" />
<esld:DisplayMultipliers />
</esld:Sld>
<esld:BusbarColoring triggeringTimeBuffer="0" />
</Private>

<Header id="ID" version="2003" revision="A" nameStructure="IEDName" />

<Communication>
<SubNetwork name="WA1" desc="Subnetwork type="8-MMS" />
<Private type="ABBPCMInternalObjRef" fe64bc9f-6918-4694-b3c7-b8f8f4d6e778" />
<ConnectedAP iedName="AATTH01" apName="S1" />
<Address>
<P type="IP" xsi:type="tP_IP">192.168.0.10</P>
<P type="OSI-AP-Title" xsi:type="tP_OSI-AP-Title">1,3,9999,23</P>
<P type="OSI-AE-Qualifier" xsi:type="tP_OSI-AE-Qualifier">23</P>
<P type="OSI-PSEL" xsi:type="tP_OSI-PSEL">000000</P>
<P type="OSI-TSEL" xsi:type="tP_OSI-TSEL">0001</P>
<P type="IP-GATEWAY" xsi:type="tP_IP-GATEWAY">0.0.0.0</P>
<P type="IP-SUBNET" xsi:type="tP_IP-SUBNET">255.255.255.0</P>
</Address>
</ConnectedAP>
</SubNetwork>
</Communication>

<IED name="AATTH01" desc="Server" type="RTU560_2" manufacturer="ABB" configVersion="1.3"
<Private type="ABB_PCMObjectTypeName">Generic IEC61850 IED</Private>
<ConnectedAP iedName="AATTH01" apName="S1" />
<Server>
<Authentication />
<Device inst="L00">
<lnClass="LLN0" inst="" lnType="LLN0_RTU560_2_IEC61850" />
<DataSet name="StatNrmIa" desc="Status data used for event list entries." />
</Device>
</Server>
</Communication>
</IED>
</Private>
</Communication>
</Scl>
</Sld>
<FCDA idInst="LD0" prefix="SP" lnClass="GGIO" lnInst="15" doName="Ind" fc="S" />
<FCDA idInst="LD0" prefix="SP" lnClass="GGIO" lnInst="14" doName="Ind" fc="S" />
<FCDA idInst="LD0" prefix="SP" lnClass="GGIO" lnInst="13" doName="Ind" fc="S" />
<FCDA idInst="LD0" prefix="SP" lnClass="GGIO" lnInst="12" doName="Ind" fc="S" />
<FCDA idInst="LD0" prefix="SP" lnClass="GGIO" lnInst="11" doName="Ind" fc="S" />
<FCDA idInst="LD0" prefix="SP" lnClass="GGIO" lnInst="10" doName="Ind" fc="S" />
<FCDA idInst="LD0" prefix="SP" lnClass="GGIO" lnInst="9" doName="Ind" fc="ST" />
<FCDA idInst="LD0" prefix="SP" lnClass="GGIO" lnInst="8" doName="Ind" fc="ST" />

<ReportControl name="rcb_A" rptId="A" dataSet="StatNm1A" confRev="1" bufTime=":"
<TrgOp sChgs="true" chgRs="true"/>
</OptFields/>
</ReportControl>
<DOI name="NamPlt">
<DAI name="swRev">
<Val>12.0.3.0</Val>
</DAI>
</DOI>
</LN>
<LN lnClass="LPHD" inst="1" lnType="LPHD_RTU560_2_IEC61850" />
<DOI name="PhyNam">
<DAI name="swRev">
<Val>12.0.3.0</Val>
</DAI>
</DOI>
<DOI name="PhyHealth" desc="IED operable">
<DAI name="stVal">
<Val name="Ind">Private type="RTU560_IED_Health">1000</Val>
</DAI>
</DOI>
</LN>/

<LN prefix="SP" lnClass="GGIO" inst="1" lnType="ABBRTU500_2_Rev1_SPGGIO_#LIB1" />
<LN prefix="SP" lnClass="GGIO" inst="9" lnType="ABBRTU500_2_Rev1_SPGGIO_#LIB1" />
<LN prefix="SP" lnClass="GGIO" inst="7" lnType="ABBRTU500_2_Rev1_SPGGIO_#LIB1" />
<LN prefix="SP" lnClass="GGIO" inst="2" lnType="ABBRTU500_2_Rev1_SPGGIO_#LIB1" />
<LN prefix="SP" lnClass="GGIO" inst="5" lnType="ABBRTU500_2_Rev1_SPGGIO_#LIB1" />
<LN prefix="SP" lnClass="GGIO" inst="16" lnType="ABBRTU500_2_Rev1_SPGGIO_#LIB1" />
<LN prefix="SP" lnClass="GGIO" inst="15" lnType="ABBRTU500_2_Rev1_SPGGIO_#LIB1" />
<LN prefix="SP" lnClass="GGIO" inst="14" lnType="ABBRTU500_2_Rev1_SPGGIO_#LIB1" />
<LN prefix="SP" lnClass="GGIO" inst="13" lnType="ABBRTU500_2_Rev1_SPGGIO_#LIB1" />
<LN prefix="SP" lnClass="GGIO" inst="12" lnType="ABBRTU500_2_Rev1_SPGGIO_#LIB1" />
<LN prefix="SP" lnClass="GGIO" inst="11" lnType="ABBRTU500_2_Rev1_SPGGIO_#LIB1" />
<LN prefix="SP" lnClass="GGIO" inst="10" lnType="ABBRTU500_2_Rev1_SPGGIO_#LIB1" />
<LN prefix="SP" lnClass="GGIO" inst="9" lnType="ABBRTU500_2_Rev1_SPGGIO_#LIB1" />
<LN prefix="SP" lnClass="GGIO" inst="8" lnType="ABBRTU500_2_Rev1_SPGGIO_#LIB1" />
</LN>
</Device>
</Server>
</AccessPoint>
</IED>
</DataTypeTemplates>
</LNodeType id="LLN0_RTU560_2_IEC61850" lnClass="LLN0" iedType="RTU560_2">
<DO name="Mod" type="tcROMod_RTU560_2_IEC61850" />
<DO name="Beh" type="tcBeh_RTU560_2_IEC61850" />
<DO name="Health" type="tcHealth_RTU560_2_IEC61850" />
<DO name="NamPlt" type="tcLPL_LLN0_RTU560_2_IEC61850" />
<DO name="Loc" type="tcSPS_RTU560_2_IEC61850" />
</LN>
</LNNodeType>
<LNNodeType id="LPHD_RTU560_2_IEC61850" lnClass="LPHD" iedType="RTU560_2">
<DO name="PhyNam" type="tcLPL_LPHD_RTU560_2_IEC61850" />
<DO name="PhyHealth" type="tcHealth_LPHD_RTU560_2_IEC61850" />
<DO name="Proxy" type="tcSPS_RTU560_2_IEC61850" />
</LN>
</LNNodeType>
<LNNodeType id="ABBRTU500_2_Rev1_SPGGIO_#LIB1" lnClass="GGIO" iedType="RTU560_2">
<DO name="Mod" type="ABBRTU500_2_Rev1_tcROMod" />
<DO name="Beh" type="ABBRTU500_2_Rev1_tcBeh" />
<DO name="Health" type="ABBRTU500_2_Rev1_tcHealth" />
<DO name="NamPlt" type="ABBRTU500_2_Rev1_tcLPL_#LIB1" />
<DO name="Ind" type="ABBRTU500_2_Rev1_tcSPS" />
</LN>
</LNNodeType>
<DOCTYPE id="tcROMod_RTU560_2_IEC61850" cdc="INC" iedType="RTU560_2">
  <DA name="stVal" bType="Enum" valKind="RO" type="Mod" fc="ST" dchg="true">
    <Val></Val>
  </DA>
  <DA name="q" bType="Quality" valKind="RO" fc="ST" qchg="true" />
  <DA name="t" bType="Timestamp" valKind="RO" fc="ST" />
  <DA name="ctlModel" bType="Enum" valKind="R0" type="ctlModel" fc="CF">
    <Val=status-only></Val>
  </DA>
  <DA name="d" bType="VisString255" fc="DC" />
</DOCTYPE>

<DOCTYPE id="tcBeh_RTU560_2_IEC61850" cdc="INS" iedType="RTU560_2">
  <DA name="stVal" bType="Enum" valKind="RO" type="Beh" fc="ST" dchg="true">
    <Val></Val>
  </DA>
  <DA name="q" bType="Quality" valKind="RO" fc="ST" qchg="true" />
  <DA name="t" bType="Timestamp" valKind="RO" fc="ST" />
  <DA name="d" bType="VisString255" fc="DC" />
</DOCTYPE>

<DOCTYPE id="tcHealth_RTU560_2_IEC61850" cdc="INS" iedType="RTU560_2">
  <DA name="stVal" bType="Enum" type="Health" fc="ST" dchg="true">
    <Val>Alarm</Val>
  </DA>
  <DA name="q" bType="Quality" valKind="RO" fc="ST" qchg="true" />
  <DA name="t" bType="Timestamp" valKind="RO" fc="ST" />
  <DA name="d" bType="VisString255" fc="DC" />
</DOCTYPE>

<DOCTYPE id="tcLPL_LL00_RTU560_2_IEC61850" cdc="LPL" iedType="RTU560_2">
  <DA name="vendor" bType="VisString255" valKind="RO" fc="DC">
    <Val>ABB</Val>
  </DA>
  <DA name="swRev" bType="VisString255" fc="DC">
    <Val>11.4.2.0</Val>
  </DA>
  <DA name="d" bType="VisString255" valKind="RO" fc="DC">
    <Val>RTU560_Server</Val>
  </DA>
  <DA name="configRev" bType="VisString255" fc="DC">
    <Val>1.0</Val>
  </DA>
  <DA name="ldNs" bType="VisString255" fc="EX">
    <Val>IEC 61850-7-4:2003</Val>
  </DA>
</DOCTYPE>

<DOCTYPE id="tcSPS_RTU560_2_IEC61850" cdc="SPS" iedType="RTU560_2">
  <DA name="stVal" bType="BOOLEAN" fc="ST" dchg="true">
    <Val>false</Val>
  </DA>
  <DA name="q" bType="Quality" valKind="RO" fc="ST" qchg="true" />
  <DA name="t" bType="Timestamp" valKind="RO" fc="ST" />
  <DA name="d" bType="VisString255" fc="DC" />
</DOCTYPE>

<DOCTYPE id="tcDPL_RTU560_2_IEC61850" cdc="DPL" iedType="RTU560_2">
  <DA name="vendor" bType="VisString255" valKind="RO" fc="DC">
    <Val>ABB</Val>
  </DA>
  <DA name="swRev" bType="VisString255" fc="DC">
    <Val>11.4.2.0</Val>
  </DA>
  <DA name="serNum" bType="VisString255" fc="DC" />
  <DA name="model" bType="VisString255" valKind="R0" fc="DC">
    <Val>RTU560_Server</Val>
  </DA>
</DOCTYPE>

<DOCTYPE id="ABBRTU500_2_Rev1_tcROMod" cdc="INC" iedType="RTU560_2">
  <DA name="stVal" bType="Enum" valKind="RO" type="Mod" fc="ST" dchg="true">
    <Val></Val>
  </DA>
  <DA name="q" bType="Quality" valKind="RO" fc="ST" qchg="true" />
  <DA name="t" bType="Timestamp" valKind="RO" fc="ST" />
  <DA name="ctlModel" bType="Enum" valKind="R0" type="ctlModel" fc="CF">
    <Val=status-only></Val>
  </DA>
</DOCTYPE>
<DA name="d" bType="VisString255" fc="DC" />
</DOType>
<DOType id="ABBRTU500_2_Rev1_tcBeh" cdc="INS" iedType="RTU560_2">
<DA name="stVal" bType="Enum" valKind="RO" type="Beh" fc="ST" dchg="true">
<Val>on</Val>
</DA>
<DA name="q" bType="Quality" valKind="RO" fc="ST" qchg="true" />
<DA name="t" bType="Timestamp" valKind="RO" fc="ST" />
<DA name="d" bType="VisString255" fc="DC" />
</DOType>
<DOType id="ABBRTU500_2_Rev1_tcHealth" cdc="INS" iedType="RTU560_2">
<DA name="stVal" bType="Enum" type="Health" fc="ST" dchg="true">
<Val>Alarm</Val>
</DA>
<DA name="q" bType="Quality" valKind="RO" fc="ST" qchg="true" />
<DA name="t" bType="Timestamp" valKind="RO" fc="ST" />
<DA name="d" bType="VisString255" fc="DC" />
</DOType>
<DOType id="ABBRTU500_2_Rev1_tcLPL_#LIB1" cdc="LPL" iedType="RTU560_2">
<DA name="vendor" bType="VisString255" valKind="RO" fc="DC">
<Val>ABB</Val>
</DA>
<DA name="swRev" bType="VisString255" fc="DC">
<Val>12.0.3.0</Val>
</DA>
<DA name="d" bType="VisString255" valKind="RO" fc="DC">
<Val>RTU560_Server</Val>
</DA>
<DA name="configRev" bType="VisString255" fc="DC">
<Val>1</Val>
</DA>
</DOType>
<EnumType id="Mod">
<EnumVal ord="1">on</EnumVal>
<EnumVal ord="2">blocked</EnumVal>
<EnumVal ord="3">test</EnumVal>
<EnumVal ord="4">off</EnumVal>
</EnumType>
<EnumType id="ctlModel">
<EnumVal ord="0">status-only</EnumVal>
<EnumVal ord="1">direct-with-normal-security</EnumVal>
<EnumVal ord="2">sbo-with-normal-security</EnumVal>
<EnumVal ord="3">direct-with-enhanced-security</EnumVal>
<EnumVal ord="4">sbo-with-enhanced-security</EnumVal>
</EnumType>
<EnumType id="Beh">
<EnumVal ord="1">on</EnumVal>
<EnumVal ord="2">blocked</EnumVal>
<EnumVal ord="3">test</EnumVal>
<EnumVal ord="4">test/blocked</EnumVal>
<EnumVal ord="5">off</EnumVal>
</EnumType>
<EnumType id="Health">
<EnumVal ord="1">OK</EnumVal>
<EnumVal ord="2">Warning</EnumVal>
<EnumVal ord="3">Alarm</EnumVal>
</EnumType>
</DataTypeTemplates>
</SCL>
GetStatus

GetStatus messages

Description

Request which queries the device for the status of the digital input.

Response which returns the result of the request and, if 'result = OK', contains the current status for the digital input.

**IEC61850 Fields**

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>FC</th>
<th>SUB ATTRIBUTE</th>
<th>DATATYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPGGIO2.ind</td>
<td>ST</td>
<td>stVal</td>
<td>BOOLEAN</td>
<td>Current state of the digital input.</td>
</tr>
</tbody>
</table>

**Example**

Soap requests and responses sent to and from platform:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="http://www.opensmartgridplatform.org/schemas/common/2014/10"
xmlns:ns1="http://www.opensmartgridplatform.org/schemas/publiclighting/adhocmanagement/2014/"
>
  <soapenv:Header>
    <ns:ApplicationName>SoapUI</ns:ApplicationName>
    <ns:UserName>Kevin</ns:UserName>
    <ns:OrganisationIdentification>LianderNetManagement</ns:OrganisationIdentification>
    <ns:MessagePriority>9</ns:MessagePriority>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:GetStatusRequest>
      <ns1:DeviceIdentification>LichtmeterNoord</ns1:DeviceIdentification>
    </soapenv:Body>
</soapenv:Envelope>

SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:GetStatusAsyncResponse xmlns:ns2="http://www.opensmartgridplatform.org/schemas/publiclighting/adhocmanagement/2014/"
      xmlns:ns3="http://www.opensmartgridplatform.org/schemas/publiclighting/adhocmanagement/2014/"
    >
      <ns3:CorrelationUid>LianderNetManagement|||LichtmeterNoord|||20181001155341143</ns3:CorrelationUid>
      <ns3:DeviceId>LichtmeterNoord</ns3:DeviceId>
    </soapenv:Body>
</SOAP-ENV:Envelope>

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="http://www.opensmartgridplatform.org/schemas/common/2014/10"
xmlns:ns1="http://www.opensmartgridplatform.org/schemas/publiclighting/adhocmanagement/2014/"
>
  <soapenv:Header>
    <ns:ApplicationName>SoapUI</ns:ApplicationName>
    <ns:UserName>Kevin</ns:UserName>
    <ns:OrganisationIdentification>LianderNetManagement</ns:OrganisationIdentification>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:GetStatusAsyncRequest>
      <ns1:AsyncRequest>
        <ns:CorrelationUid>LianderNetManagement|||LichtmeterNoord|||20181001155341143</ns:CorrelationUid>
        <ns:DeviceId>LichtmeterNoord</ns:DeviceId>
      </ns1:AsyncRequest>
    </soapenv:Body>
</soapenv:Envelope>

SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:GetStatusResponse xmlns:ns2="http://www.opensmartgridplatform.org/schemas/publiclighting/adhocmanagement/2014/"
      xmlns:ns3="http://www.opensmartgridplatform.org/schemas/publiclighting/adhocmanagement/2014/"
    >
      <ns3:Result>OK</ns3:Result>
    </soapenv:Body>
</SOAP-ENV:Envelope>
```
Platform message of the data read from the device:

LogicalDevice: SWDeviceGenericIO
messageType: GetLightSensorStatus
  SPGGIO2.Ind[ST].stVal: false

IEC61850 protocol Adapter logging:

2018-10-01 15:53:41.468 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-23] INFO o.o.a.p.i.i.m.DeviceRequestMessageListener@onMessage:61 - Received message of type: GET_LIGHT_SENSOR_STATUS with message priority: 9
2018-10-01 15:53:41.468 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-23] INFO o.o.a.p.i.i.m.BaseMessageProcessor@printDomainInfo:53 - Calling DeviceService function: GET_LIGHT_STATUS for domain: PUBLIC_LIGHTING 1.0
2018-10-01 15:53:41.468 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-23] INFO o.o.a.p.i.i.m.BaseMessageProcessor@getJmsXdeliveryCount:64 - jmsXdeliveryCount: 1
2018-10-01 15:53:41.469 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-23] INFO o.o.a.p.i.i.n.s.Iec61850DeviceConnectionService@testIfConnectionIsCachedAndAlive:206 - Connection found for deviceIdentification: LichtmeterNoord
2018-10-01 15:53:41.469 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-23] INFO o.o.a.p.i.i.n.s.Iec61850DeviceConnectionService@testIfConnectionIsCachedAndAlive:224 - Connection is still active for deviceIdentification: LichtmeterNoord
2018-10-01 15:53:41.472 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-23] INFO o.o.a.p.i.i.n.h.DeviceConnection@createObjectReference:94 - Device: LichtmeterNoord, ObjectReference: AA1TH01LD0/SPGGIO2.Ind
2018-10-01 15:53:41.473 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-23] INFO o.o.a.p.i.i.n.s.c.Iec61850GetLightSensorStatusCommand@apply:89 - device: LichtmeterNoord, indNode: AA1TH01LD0/SPGGIO2.Ind [ST]
AA1TH01LD0/SPGGIO2.Ind.stVal: false
AA1TH01LD0/SPGGIO2.Ind.q: 0x00 0x00
AA1TH01LD0/SPGGIO2.Ind.t: Thu Jan 01 00:00:00 UTC 1970
2018-10-01 15:53:41.473 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-23] INFO o.o.a.p.i.i.n.s.c.Iec61850GetLightSensorStatusCommand@apply:93 - device: LichtmeterNoord, stVal: AA1TH01LD0/SPGGIO2.Ind.stVal: false
2018-10-01 15:53:41.474 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-23] INFO o.o.a.p.i.i.m.LmdDeviceRequestMessageProcessor@handleGetStatusDeviceResponse:65 - Handling getStatusDeviceResponse for device: LichtmeterNoord
2018-10-01 15:53:41.474 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-23] INFO o.o.a.p.i.i.m.LmdDeviceRequestMessageProcessor@handleGetStatusDeviceResponse:65 - Handling getStatusDeviceResponse for device: LichtmeterNoord
2018-10-01 15:53:41.474 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-23] INFO o.o.a.p.i.i.n.s.Iec61850LmdDeviceService@enableReporting:125 - Trying to enable reporting for device: LichtmeterNoord
2018-10-01 15:53:41.474 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-23] INFO o.o.a.p.i.i.n.h.DeviceConnection@createObjectReference:94 - Device: LichtmeterNoord, ObjectReference: AA1TH01LD0/LLN0.rcb_A
2018-10-01 15:53:41.474 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-23] INFO o.o.a.p.i.i.n.s.c.Iec61850EnableReportingCommand@enableUnbufferedReportingOnLightMeasurementDevice:146 - reportingEnabled for unbuffered reports: true
2018-10-01 15:53:41.474 [osgp-tst-03] [iec61850RequestsMessageListenerContainer-23] INFO o.o.a.p.i.i.n.s.c.Iec61850EnableReportingCommand@enableUnbufferedReportingOnLightMeasurementDevice:146 - reportingEnabled for unbuffered reports: true
AA1TH01LD0/SPGGIO2.Ind.stVal: false
AA1TH01LD0/SPGGIO2.Ind.q: 0x00 0x00
AA1TH01LD0/SPGGIO2.Ind.t: Thu Jan 01 00:00:00 UTC 1970

EventNotification

EventNotification messages

Description

A light sensor device can send a buffered report containing the current state of the digital inputs. The buffered report will be sent on data change. OSGP will interpret the buffered report and save event information contained in the report, in this case the state of the digital input:

AA1TH01LD0/SPGGIO1.Ind.stVal: false

IEC61850 protocol adapter logging:

2018-10-01 16:11:34.057] [osgp-tst-03] [Thread-613] INFO o.o.a.p.i.i.n.r.Iec61850ClientLMDE
2018-10-01 16:11:34.057] [Thread-613] INFO o.o.a.p.i.i.n.r.Iec61850ClientLMDE
RptId: A
DataSetRef: AA1TH01LD0/LLN0.StatNrm1A
ConfRev: null
BufOvfl: null
EntryId: null
InclusionBitString: [false, true, false, false, true, false, false, false, MoreSegmentsFollow: false
SnNum: null
SubSqNum: null
TimeOfEntry: null
DataSet: AA1TH01LD0/LLN0.StatNrm1A
DataSet members: 4
member: AA1TH01LD0/SPGGIO1.Ind [ST]
AA1TH01LD0/SPGGIO1.Ind.stVal: false
AA1TH01LD0/SPGGIO1.Ind.q: 0x00 0x00
AA1TH01LD0/SPGGIO1.Ind.t: Thu Jan 01 00:00:00 UTC 1970
member: AA1TH01LD0/SPGGIO2.Ind [ST]
AA1TH01LD0/SPGGIO2.Ind.stVal: true
AA1TH01LD0/SPGGIO2.Ind.q: 0x00 0x00
AA1TH01LD0/SPGGIO2.Ind.t: Thu Jan 01 00:00:00 UTC 1970
member: AA1TH01LD0/SPGGIO3.Ind [ST]
AA1TH01LD0/SPGGIO3.Ind.stVal: false
AA1TH01LD0/SPGGIO3.Ind.q: 0x00 0x00
AA1TH01LD0/SPGGIO3.Ind.t: Thu Jan 01 00:00:00 UTC 1970
member: AA1TH01LD0/SPGGIO4.Ind [ST]
AA1TH01LD0/SPGGIO4.Ind.stVal: false
AA1TH01LD0/SPGGIO4.Ind.q: 0x00 0x00
AA1TH01LD0/SPGGIO4.Ind.t: Thu Jan 01 00:00:00 UTC 1970

2018-10-01 16:11:34.057] [osgp-tst-03] [Thread-613] INFO o.o.a.p.i.i.n.r.Iec61850ClientLMDE
2018-10-01 16:11:34.125] [osgp-tst-03] [Thread-613] INFO o.o.a.p.i.i.n.r.Iec61850ClientLMDE
2018-10-01 16:11:34.125] [Thread-613] INFO o.o.a.p.i.i.n.r.Iec61850ClientLMDE
2018-10-01 16:11:34.139] [Thread-613] INFO o.o.a.p.i.a.s.DeviceManagementService
2018-10-01 16:11:34.139] [Thread-613] INFO o.o.a.p.i.a.s.DeviceManagementService
2018-10-01 16:11:34.139] [Thread-613] INFO o.o.a.p.i.a.s.DeviceManagementService
2018-10-01 16:11:34.145] [osgp-tst-03] [ActiveMQ Task-1] INFO o.a.a.t.failover.Failover.Trans:
2018-10-01 16:11:34.198] [osgp-tst-03] [Thread-613] INFO o.o.a.p.i.a.s.DeviceManagementService
2018-10-01 16:11:34.198] [Thread-613] INFO o.o.a.p.i.a.s.DeviceManagementService
2018-10-01 16:11:34.198] [Thread-613] INFO o.o.a.p.i.a.s.DeviceManagementService
2018-10-01 16:11:34.198] [Thread-613] INFO o.o.a.p.i.a.s.DeviceManagementService
2018-10-01 16:11:34.198] [Thread-613] INFO o.o.a.p.i.a.s.DeviceManagementService
2018-10-01 16:11:34.198] [Thread-613] INFO o.o.a.p.i.a.s.DeviceManagementService

FlexOVL_540_171101_2 433
**DLMS / COSEM**

**DLMS/COSEM**

The open smart grid platform supports DLMS/COSEM (IEC 62056). DLMS/COSEM is a popular protocol to read smart meters. DLMS/COSEM is the de facto standard in Europe.

The open smart grid platform DLMS/COSEM implementation was initial based on SMR5 and DSMR v4. Other types of meters/profiles can be added to the platform. The open smart grid platform implementation supports HLS3/4/5.

**Protocol security**

- Public/private key pair(s)
- Multiple encryption levels inside protocol (DSMR requires highest encryption level)
- Full encryption of communication

**Used library**

The OpenMUC jDLMS library from Fraunhofer is used to implement the protocol. Please note that jDLMS is licensed under the GPLv3.

**DLMS device simulator**

[DLMS device simulator]

**Supported devices**

These devices can be used in combination with the Open Smart Grid Platform.

<table>
<thead>
<tr>
<th>E/G</th>
<th>Supplier</th>
<th>Type</th>
<th>x fase</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>Kaifa</td>
<td>MA105</td>
<td>1 fase</td>
</tr>
<tr>
<td>E</td>
<td>Kaifa</td>
<td>MA105C</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>Kaifa</td>
<td>MA104</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>Kaifa</td>
<td>MA304</td>
<td>3 fase</td>
</tr>
<tr>
<td>E</td>
<td>L+G</td>
<td>ZCF110CCtFs2</td>
<td>1 fase</td>
</tr>
<tr>
<td>E</td>
<td>L+G</td>
<td>ZMF110CCtFs2</td>
<td>3 fase</td>
</tr>
<tr>
<td>E</td>
<td>L+G</td>
<td>ZCF110CBtFs2</td>
<td>1 fase</td>
</tr>
<tr>
<td>E</td>
<td>L+G</td>
<td>E350</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>L+G</td>
<td>E650</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>Iskra</td>
<td>Mx382</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>L+G</td>
<td>G350 (wireless/wired)</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>Itron</td>
<td>G1 RF1</td>
<td></td>
</tr>
</tbody>
</table>
DLMS device simulator

**DLMS device simulator**

The library that is used to connect to DLMS devices contains functionality to build a simulator for a device. The library offers the following core functionality:

- zero or more servers can be started on a host (different ports)
- zero or more logical devices can be registered with a server (different device id)
- zero or more annotated objects can be registered with a logical device
- these objects define available dlms classes, ObisCodes, attributeIds and methods for the device and can contain any logic
- authentication and encryption are supported

If you want to simulate a certain device you will prepare annotated classes and register instances of these with a logical device. Because you create plain Java you can make use of all functionality Java offers, for example databases. To try and make the simulation more realistic you may build in connection timeouts etc.

**Usage**

For each combination of a cosem class and obiscode you create a Java class that you annotate with `@CosemClass(id = ..., obis = "x.x.x.x.255")`

In these Java classes you can add fields of type DataObject that you annotate with `@CosemAttribute(id = ..., type = Type.x)`

Also you can create getXXX and setXXX methods to intercept getting and setting data on a logical device. XXX will be the name of the corresponding field starting with a capital letter.

For example:

```java
@CosemClass(id = 3, obis = "1.0.1.8.0.255")
public class ImportValue {
    @CosemAttribute(id = 2, type = Type.DOUBLE_LONG_UNSIGNED)
    private DataObject d1 = DataObject.newUInteger32Data(10001);

    @CosemAttribute(id = 3, type = Type.STRUCTURE)
    private DataObject d2 = DataObject.newStructureData(DataObject.newInteger8Data((byte) -2),
                DataObject.newInteger8Data((byte) 30));

    public void setD1(DataObject newData) throws IllegalAttributeAccessException {
        // ....
    }

    public DataObject getD1() throws IllegalAttributeAccessException {
        return d1;
    }
}
```

The value of the field will be the response to `get(AttributeAddress...)` that is fired from osgp CommandExecutors. NOTE that these values can also be set! For example using the ClientConsole.

You can also annotate methods with or without a DataObject return value and with or without a DataObject parameter: `@CosemMethod(id = ..., consumes = Type.x)`

For example:

```java
@CosemMethod(id = 1)
public void hello() throws IllegalMethodAccessException {
    System.out.println("Has been called");
    return;
}

@CosemMethod(id = 2, consumes = Type.OCTET_STRING)
public DataObject hello2(DataObject data) throws IllegalMethodAccessException {
    throw new IllegalMethodAccessException(MethodResultCode.OTHER_REASON);
}
```

Such a method will be called when osgp fires ClientConnection.action, the DataObject that may be returned will become available in osgp on the MethodResult object.

**Per command design**
The way to implement request/response for a command in the simulator is as follows.

- create a separate package per command
- create a java class for each combination of classid and obiscode
- in this class implement the attribute id's and the methods that will be requested as explained above
OSLP

OSLP Documentation

The Open Street Light Protocol

The OSLP is a lightweight message based protocol. OSLP uses Google Protocol Buffers and is used for communication with SSLD devices (and device simulators). It is defined as a contract/interface. The interface defines datatypes and messages which use those data types. Google Protocol Buffers is used to generate the protocol implementations for Java (for the platform) and C/C++ (for the SSLD devices).

Open street light protocol does not use ASN.1 but Google Protocol Buffers. The main reason for this is the lack of a good quality free ASN.1 compiler for Java or C. Google Protocol Buffers offers a fast and free compiler for Java and C which produces small message sizes.

Protocol security

- Public/private key pair
- Signing of messages through Elliptic Curve DSA 256 bit Integrity of the message is ensured Sender identity is ensured ** No encryption, because content is not confidential
- Replay attack prevention

Special note on Java security provider:

When both the DLMS and OSLP providers are deployed within the same Java VM, the SunEC provider will not work properly. To workaround this issue, the SunPKCS11-NSS provider must be used for the OSLP protocol adapter. By default this provider is enabled on the development VM.

OSLP v0.5.1 (Deprecated)

The protobuf contract for OSLP v0.5.1. For v0.5.1 port number 12121 is used.

OSLP v0.6.1

The protobuf contract for OSLP v0.6.1. For v0.6.1 port number 12122 is used.

OSLP Envelope

The requests and responses are sent using an OSLP envelope. This structure contains the following fields: securityKey, sequenceNumber, deviceId and payloadMessage. The first 3 field are byte arrays, the payloadMessage is a protobuf type which is serializable.

```java
class OslpEnvelope {
    /**
     * Length of the security hash.
     * Length for ECDSA is 71 or 72 or 73 bytes.
     * Length for RSA is 128 bytes.
     */
    public static final int SECURITY_KEY_LENGTH = 128;

    /**
     * Length of the sequence number.
     */
    public static final int SEQUENCE_NUMBER_LENGTH = 2;

    /**
     * Length of the manufacturer id.
     */
    public static final int MANUFACTURER_ID_LENGTH = 2;

    /**
     * Length of the device id.
     */
    public static final int DEVICE_ID_LENGTH = 10;

    /**
     * Length of the length.
     */
    public static final int LENGTH_INDICATOR_LENGTH = 2;
}
```
/**
 * Buffer for security key bytes.
 */
public byte[] securityKey = new byte[SECURITY_KEY_LENGTH];

/**
 * Buffer for sequence number bytes.
 */
public byte[] sequenceNumber = new byte[SEQUENCE_NUMBER_LENGTH];

/**
 * Buffer for deviceid bytes.
 */
public byte[] deviceId = new byte[DEVICE_ID_LENGTH + MANUFACTURER_ID_LENGTH];

/**
 * Buffer for OSLP payload.
 */
public Message payloadMessage;
OSLP v0.5.1

Contract

NOTE: OSLP v0.5.1 is deprecated.

Contract for v0.5.1
Protobuf Contract

OSLP protobuf file, v0.5.1

// import "nanopb.proto";

package oslp;

option java_package = "org.opensmartgridplatform.oslp";

message Message {
  optional RegisterDeviceRequest registerDeviceRequest = 1;
  optional RegisterDeviceResponse registerDeviceResponse = 2;
  optional StartSelfTestRequest startSelfTestRequest = 3;
  optional StartSelfTestResponse startSelfTestResponse = 4;
  optional StopSelfTestRequest stopSelfTestRequest = 5;
  optional StopSelfTestResponse stopSelfTestResponse = 6;
  optional UpdateFirmwareRequest updateFirmwareRequest = 7;
  optional UpdateFirmwareResponse updateFirmwareResponse = 8;
  optional SetLightRequest setLightRequest = 9;
  optional SetLightResponse setLightResponse = 10;
  optional GetStatusRequest getStatusRequest = 11;
  optional GetStatusResponse getStatusResponse = 12;
  optional ResumeScheduleRequest resumeScheduleRequest = 13;
  optional ResumeScheduleResponse resumeScheduleResponse = 14;
  optional SetEventNotificationsRequest setEventNotificationsRequest = 15;
  optional SetEventNotificationsResponse setEventNotificationsResponse = 16;
  optional EventNotificationRequest eventNotificationRequest = 17;
  optional EventNotificationResponse eventNotificationResponse = 18;
  optional GetFirmwareVersionRequest getFirmwareVersionRequest = 19;
  optional GetFirmwareVersionResponse getFirmwareVersionResponse = 20;
  optional SetScheduleRequest setScheduleRequest = 21;
  optional SetScheduleResponse setScheduleResponse = 22;
  optional SetConfigurationRequest setConfigurationRequest = 25;
  optional SetConfigurationResponse setConfigurationResponse = 26;
  optional GetPowerUsageHistoryRequest getPowerUsageHistoryRequest = 27;
  optional GetPowerUsageHistoryResponse getPowerUsageHistoryResponse = 28;
  optional GetActualPowerUsageRequest getActualPowerUsageRequest = 29;
  optional GetActualPowerUsageResponse getActualPowerUsageResponse = 30;
  optional SetRebootRequest setRebootRequest = 31;
  optional SetRebootResponse setRebootResponse = 32;
  optional SetTransitionRequest setTransitionRequest = 33;
  optional SetTransitionResponse setTransitionResponse = 34;
  optional GetConfigurationRequest getConfigurationRequest = 35;
  optional GetConfigurationResponse getConfigurationResponse = 36;
  optional ConfirmRegisterDeviceRequest confirmRegisterDeviceRequest = 37;
  optional ConfirmRegisterDeviceResponse confirmRegisterDeviceResponse = 38;
}

// ========= Device Installation
message RegisterDeviceRequest {
  required string deviceIdentification = 1; // [(nanopb).max_size = 41];
  required bytes ipAddress = 2; // [(nanopb).max_size = 4];
  required DeviceType deviceType = 3;
  required bool hasSchedule = 4;
  required uint32 randomDevice = 5; // 16 bits
}

message RegisterDeviceResponse {
  required Status status = 1;
  required string currentTime = 2; // [(nanopb).max_size = 15];// - format YYYYMMDDhhmmss
  required uint32 randomDevice = 3;
  required uint32 randomPlatform = 4;
  optional LocationInfo locationInfo = 5; // Location information of device
}

message StartSelfTestRequest {
  optional bool present = 1 [default = true];
}

message StartSelfTestResponse {

message StopSelfTestRequest {
  optional bool present = 1 [default = true];
}

message StopSelfTestResponse {
  required Status status = 1;
  required bytes selfTestResult = 2 ;
}

// ========= Firmware Management
message GetFirmwareVersionRequest {
  optional bool present = 1 [default = true];
}

message GetFirmwareVersionResponse {
  required string firmwareVersion = 1 ;
}

message UpdateFirmwareRequest {
  required string firmwareDomain = 1 ;
  required string firmwareUrl = 2 ;
}

message UpdateFirmwareResponse {
  required Status status = 1 ;
}

// ========= Ad-Hoc & Status
message SetLightRequest {
  repeated LightValue values = 1 ;
}

message SetLightResponse {
  required Status status = 1 ;
}

message GetStatusRequest {
  optional bool present = 1 [default = true];
}

message GetStatusResponse {
  required Status status = 1 ;
  repeated LightValue value = 2 ;
  required LinkType preferredLinktype = 3 ;
  required LinkType actualLinktype = 4 ;
  required LightType lightType = 5 ;
  required uint32 eventNotificationMask = 6 ;
}

message ResumeScheduleRequest {
  optional bytes index = 1 ;
  required bool immediate = 2 ;
}

message ResumeScheduleResponse {
  required Status status = 1 ;
}

message SetRebootRequest {
  optional bool present = 1 [default = true];
}

message SetRebootResponse {
  required Status status = 1 ;
}

message SetTransitionRequest {
  required TransitionType transitionType = 1 ;
  optional string time = 2 ;
}

OSLP v0.5.1
message SetTransitionResponse {
  required Status status = 1;
}

message SetEventNotificationsRequest {
  required uint32 NotificationMask = 1; // Bitmask for max 32 events, using NotificationBi
}

message SetEventNotificationsResponse {
  required Status status = 1;
}

message EventNotificationRequest {
  repeated EventNotification notifications = 1; // [(nanopb).max_count = 6];
}

message EventNotificationResponse {
  required Status status = 1;
}

// ========= Scheduling
message SetScheduleRequest {
  repeated Schedule schedules = 1; // [(nanopb).max_count = 50];
  optional PageInfo pageInfo = 2;
  required RelayType scheduleType = 3; // RT_NOT_SET is NOT supported!
}

message SetScheduleResponse {
  required Status status = 1;
}

// ========= Configuration
message SetConfigurationRequest {
  optional LightType lightType = 1;
  optional DaliConfiguration daliConfiguration = 2; // contains specific configuration for
  optional RelayConfiguration relayConfiguration = 3; // contains specific configuration if
  optional uint32 shortTermHistoryIntervalMinutes = 4;
  optional LinkType preferredLinkType = 5;
  optional MeterType meterType = 6;
  optional uint32 longTermHistoryInterval = 7;
  optional LongTermIntervalType longTermHistoryIntervalType = 8;
}

message SetConfigurationResponse {
  required Status status = 1;
}

message GetConfigurationRequest {
  optional bool present = 1 [default = true];
}

message GetConfigurationResponse {
  required Status status = 1;
  optional LightType lightType = 2;
  optional DaliConfiguration daliConfiguration = 3; // contains specific configuration for
  optional RelayConfiguration relayConfiguration = 4; // contains specific configuration if
  optional uint32 shortTermHistoryIntervalMinutes = 5;
  optional LinkType preferredLinkType = 6;
  optional MeterType meterType = 7;
  optional uint32 longTermHistoryInterval = 8;
  optional LongTermIntervalType longTermHistoryIntervalType = 9;
}

message ConfirmRegisterDeviceRequest {
  required uint32 randomDevice = 1;
  required uint32 randomPlatform = 2;
}

message ConfirmRegisterDeviceResponse {
  required Status status = 1;
}
required uint32 randomDevice = 2;  
required uint32 randomPlatform = 3;  
required uint32 sequenceWindow = 4;  

// ========= Monitoring
message GetPowerUsageHistoryRequest {  
  required TimePeriod timePeriod = 1;  
  optional uint32 page = 2;  
  required HistoryTermType termType = 3;  
}

message GetPowerUsageHistoryResponse {  
  required Status status = 1;  
  repeated PowerUsageData powerUsageData = 2;  
  // [(nanopb).max_count = 20];  
  optional PageInfo pageInfo = 3;  
}

message GetActualPowerUsageRequest {  
  optional bool present = 1 [default = true];  
}

message GetActualPowerUsageResponse {  
  required Status status = 1;  
  required PowerUsageData powerUsageData = 2;  
}

// ========= Types
message LocationInfo {  
  optional sint32 timeOffset = 1;  // correction in minutes with respect to UTC  
  optional sint32 latitude = 2;  // divide by 1000000 to get float value  
  optional sint32 longitude = 3;  // divide by 1000000 to get float value  
}

message LightValue {  
  optional bytes index = 1;  // [(nanopb).max_size = 1];  // index number of connected light  
  required bool on = 2;  
  optional bytes dimValue = 3;  // [(nanopb).max_size = 1];  // 1 - 100 %  
}

message EventNotification {  
  required Event event = 1;  
  optional bytes index = 2;  // [(nanopb).max_size=1];  
  optional string description = 3;  // [(nanopb).max_size = 81];  
}

message Schedule {  
  required Weekday weekday = 1;  
  optional string startDay = 2;  // [(nanopb).max_size = 9];  // - format YYYYMMDD UTC, indic  
  optional string endDay = 3;  // [(nanopb).max_size = 9];  // - format YYYYMMDD UTC, includ  
  required ActionTime actionTime = 4;  
  optional string time = 5;  // [(nanopb).max_size = 7];  // - format hhmmss localtime set w  
  repeated LightValue value = 7;  // [(nanopb).max_count = 6];  
  optional TriggerType triggerType = 8;  // React to setTransition or switch astronomical  
}

message Window {  
  required uint32 minutesBefore = 1;  // minutes before sunset / sunrise  
  required uint32 minutesAfter = 2;  // minutes after sunset / sunrise  
}

message DaliConfiguration {  
  optional bytes numberOfLights = 1;  // [(nanopb).max_size = 1];  // number of lights conne  
  repeated IndexAddressMap addressMap = 2;  // [(nanopb).max_count = 4];  
}

message RelayConfiguration {  
  repeated IndexAddressMap addressMap = 1;  // [(nanopb).max_count = 6];  
}

message IndexAddressMap {  
  Container -
message PageInfo {
  required uint32 currentPage = 1;  // Pages start from 1
  required uint32 pageSize = 2;
  required uint32 totalPages = 3;
}

message TimePeriod {
  required string startTime = 1;  // - format YYYYMMDDhhmm
  required string endTime = 2;  // - format YYYYMMDDhhmmss U
}

message PowerUsageData {
  required string recordTime = 1;  // Record time - format YYYYMMDDhhmmss U
  required MeterType meterType = 2;  // Meter type (P1, Pulse, Aux)
  required uint64 totalConsumedEnergy = 3;  // Electricity delivered
  required uint32 actualConsumedPower = 4;  // Actual Electricity power delivered in W
  optional PsldData psldData = 5;
  optional SsldData ssldData = 6;
}

message PsldData {
  required uint32 totalLightingHours = 1;  // Total lighting hours
}

message SsldData {
  required uint32 totalLightingMinutes = 1;  // Total lighting minutes for lighting relay
}

message RelayData {
  required bytes index = 1;  // external index, for example 1
  required uint32 totalLightingMinutes = 2;  // Total lighting minutes for lighting relay
}

enum NotificationBit {
  DIAG_EVENTS = 1;
  HARDWARE_FAILURE = 2;
  LIGHT_EVENTS = 4;  // For example LightValue changes
  TARIFF_EVENTS = 8;  // For example Tariff changes
  MONITOR_EVENTS = 16;  // For example monitor buffer is almost full
  FIRMWARE_EVENTS = 32;  // For example firmware activation
  COMM_EVENTS = 64;  // For example alternative channel
  SECURITY_EVENTS = 128;  // For example out of sequence
}

// Events must map to their notification bit:
//EG: 0000-0999 =1
// 1000-1999 =2
// 2000-2999 =4
// 3000-3999 =8
// 4000-4999 =16
// 5000-5999 =32
// 6000-6999 =64
// 7000-7999 =128
// OR to check 2^((event num)/1000)=notification bit
enum Event {
    // 0 - 999 Diagnostics
    DIAG_EVENTS_GENERAL = 0;

    // 1000 - 1999 Hardware Failures
    HARDWARE_FAILURE_RELAY = 1000; // Index indicates relay (not supported yet)

    // 2000 - 2999 Light Events
    LIGHT_EVENTS_LIGHT_ON = 2000; // Index indicates light
    LIGHT_EVENTS_LIGHT_OFF = 2001; // Index indicates light
    LIGHT_FAILURE_DALI_COMMUNICATION = 2500; // DALI communication failure
    LIGHT_FAILURE_BALLAST = 2501; // Ballast failure detected (DALI only)
    LIGHT_FAILURE_TARIFF_SWITCH_ATTEMPT = 2502; // Attempt to switch an endpoint configured

    // 3000 - 3999 Tariff Events
    TARIFF_EVENTS_TARIFF_ON = 3000; // Tariff switched on
    TARIFF_EVENTS_TARIFF_OFF = 3001; // Tariff switched off

    // 4000 - 4999
    MONITOR_EVENTS_LONG_BUFFER_FULL = 4000; // Long term monitoring buffer overrun occurred
    MONITOR_FAILURE_P1_COMMUNICATION = 4500; // P1 meter could not be read
    MONITOR_SHORT_DETECTED = 4600;
    MONITOR_SHORT_RESOLVED = 4601;
    MONITOR_DOOR_OPENED = 4700;
    MONITOR_DOOR_CLOSED = 4701;

    // 5000 - 5999 Firmware Events
    FIRMWARE_EVENTS_ACTIVATING = 5000; // Start activating new firmware, after downloading
    FIRMWARE_EVENTS_DOWNLOAD_NOTFOUND = 5501; // Download of firmware failed, i.e. location incorrect
    FIRMWARE_EVENTS_DOWNLOAD_FAILED = 5502; // Download of firmware failed, image incorrect

    // 6000 - 6999
    COMM_EVENTS_ALTERNATIVE_CHANNEL = 6000; // Alternative channel selected for communication
    COMM_EVENTS_RECOVERED_CHANNEL = 6001; // Communication has been recovered for this channel

    // 7000 - 7999
    SECURITY_EVENTS_OUT_OF_SEQUENCE = 7000; // Out of sequence occurred and sequence number
}

// ========= Enums
enum TriggerType {
    TT_NOT_SET = 0;
    LIGHT_TRIGGER = 1;
    ASTRONOMICAL = 2;
}

enum TransitionType {
    NIGHT_DAY = 0;
    DAY_NIGHT = 1;
}

enum Weekday {
    MONDAY = 1;
    TUESDAY = 2;
    WEDNESDAY = 3;
    THURSDAY = 4;
    FRIDAY = 5;
    SATURDAY = 6;
    SUNDAY = 7;
    WEEKDAY = 8;
    WEEKEND = 9;
    ABSOLUTEDAY = 10;
    ALL = 11;
}

enum ActionTime {
    ABSOLUTETIME = 1;
    SUNRISE = 2;
    SUNSET = 3;
}
enum DeviceType {
    PSLD = 0;
    SSLD = 1;
}

enum Status {
    OK = 0;
    FAILURE = 1; // general failure
    REJECTED = 2; // request received in wrong state
}

enum LightType {
    LT_NOT_SET = 0;
    RELAY = 1;
    ONE_TO_TEN_VOLT = 2;
    ONE_TO_TEN_VOLT.Reverse = 3;
    DALI = 4;
}

enum RelayType {
    RT_NOT_SET = 0;
    LIGHT = 1;
    TARIFF = 2;
}

enum MeterType {
    MT_NOT_SET = 0;
    P1 = 1;
    PULSE = 2;
    AUX = 3;
}

enum LinkType {
    LINK_NOT_SET = 0;
    GPRS = 1;
    CDMA = 2;
    ETHERNET = 3;
}

enum LongTermIntervalType {
    LT_INT_NOT_SET = 0;
    DAYS = 1;
    MONTHS = 2;
}

enum HistoryTermType {
    Short = 0;
    Long = 1;
}
OSLP v0.6.1

Contract

Contract for v0.6.1. The contract specifies the messages which can be exchanged with an SSLD.

Messages

These messages below are part of OSLP v0.6.1. Note that OSLP v0.6.1 is backwards compatible with OSLP v0.5.1. Therefore, v0.6.1 offers the same RegisterDeviceRequest as v0.5.1 for example.

- **RegisterDeviceRequest** (from device to platform) is a request that notifies the platform a device which wants to register. During the registration the sequence number is reset to a random value, the platform is notified if the device has a light schedule, the type of the device, the device identification, and the device communicates its IP address to the platform.

- **RegisterDeviceResponse** (from platform to device) is a response which holds the time of the platform so the device can synchronize the time, contains location information for the device like GPS coordinates and Day Light Saving time information. The device will send ConfirmRegisterDeviceRequest after receiving the RegisterDeviceResponse.

- **ConfirmRegisterDeviceRequest** (from device to platform) is a request that notifies the platform that a device wants to perform the second step of the registration process.

- **ConfirmRegisterDeviceResponse** (from platform to device) is a response which confirms the ConfirmRegisterDeviceRequest has been executed or rejected.

- **StartSelfTestRequest** (from platform to device) is a request that notifies the device to switch all relays on.

- **StartSelfTestResponse** (from device to platform) is a response which confirms the StartSelfTestRequest has been executed or rejected.

- **StopSelfTestRequest** (from platform to device) is a request that notifies the device to switch all relays off.

- **StopSelfTestResponse** (from device to platform) is a response which confirms the StopSelfTestRequest has been executed or rejected.

- **UpdateFirmwareRequest** (from platform to device) is a request which notifies the device to download a new firmware version from a server using a URL.

- **UpdateFirmwareResponse** (from device to platform) is a response which confirms the UpdateFirmwareRequest has been executed or rejects the UpdateFirmwareRequest. Please note there are several events which are sent from the device to the platform to inform the platform when the firmware has been downloaded and whether or not the firmware was successfully activated.

- **SetLightRequest** (from platform to device) is a request that notifies the device to switch on or off one or several light relays, optionally with a dim-value per relay.

- **SetLightResponse** (from device to platform) is a response which confirms the SetLightRequest has been executed or rejected.

- **GetStatusRequest** (from platform to device) is a request that requires the device to send the status of all relays, current network link and preferred network link, the type of configuration (PSLD vs SSLD), and the event notification mask which has been set.

- **GetStatusResponse** (from device to platform) is a response which confirms the GetStatusRequest has been executed and returns the current status for all of the relays and other information or rejects the GetStatusRequest.

- **ResumeScheduleRequest** (from platform to device) is a request that notifies the device to continue the current schedule after the current schedule was interrupted (for example by switching by hand using SetLightRequest). This request can operate on a single relay or on all relays and the resuming of the schedule can be immediate or at the next schedule-entry.

- **ResumeScheduleResponse** (from device to platform) is a response which confirms the ResumeScheduleRequest has been executed or rejected.

- **SetEventNotificationsRequest** (from platform to device) is a request that sets the event notification mask.

- **SetEventNotificationsResponse** (from device to platform) is a response which confirms the SetEventNotifications request has been executed or rejected.

- **EventNotificationRequest** (from device to platform) is a request that pushes an event notification from a device to the platform.
- **EventNotificationResponse** (from platform to device) is a response which confirms the EventNotificationRequest has been executed or rejected.

- **GetFirmwareVersionRequest** (from platform to device) is a request that requests the device to send its current firmware version.

- **GetFirmwareVersionResponse** (from device to platform) is a response that sends the current firmware version to the platform.

- **SetScheduleRequest** (from platform to device) is a request that sends a light or tariff schedule to the device.

- **SetScheduleResponse** (from device to platform) is a response which confirms the SetScheduleRequest has been executed or rejected.

- **SetConfigurationRequest** (from platform to device) is a request that sends configuration settings to the device.

- **SetConfigurationResponse** (from device to platform) is a response which confirms the SetConfigurationRequest has been executed or rejected.

- **GetConfigurationRequest** (from platform to device) is a request that requests the device to send its current configuration settings.

- **GetConfigurationResponse** (from device to platform) is a response which confirms the GetConfigurationRequest has been executed or rejected.

- **SetRebootRequest** (from platform to device) is a request that notifies the device to reboot immediately.

- **SetRebootResponse** (from device to platform) is a response which confirms the SetRebootRequest has been executed or rejected.

- **SetTransitionRequest** (from platform to device) is a request that notifies the device to switch its light relays according to light measurement schedule-entries.

- **SetTransitionResponse** (from device to platform) is a response which confirms the SetTransitionRequest has been executed or rejected.

- **UpdateDeviceSslCertification** (from platform to device) is a request which commands a device to download a new certificate file from a server using a URL.

- **UpdateDeviceSslCertification** (from platform to device) is a response which returns the result of the UpdateFirmwareRequest. Please note there are several events which are sent from the device to the platform to inform the platform whether or not the certificate file was successfully downloaded and activated.

- **SetDeviceVerificationKeyRequest** (from platform to device) is a request which sends a new OSGP public key to the device.

- **SetDeviceVerificationKeyResponse** (from platform to device) is a response which returns the result of the SetDeviceVerificationKeyRequest.

- **SwitchFirmwareRequest** (from platform to device) is a request which commands the device to switch to the other firmware bank.

- **SwitchFirmwareResponse** (from platform to device) is a response which returns the result of the SwitchFirmwareRequest.

- **SwitchConfigurationRequest** (from platform to device) is a request which commands the device to switch to the other configuration bank.

- **SwitchConfigurationResponse** (from platform to device) is a response which returns the result of the SwitchConfigurationRequest.
 Protobuf Contract

**OSLP protobuf file, v0.6.1**

```protobuf
// import "nanopb.proto";

package oslp;

option java_package = "org.opensmartgridplatform.oslp";

message Message {
  optional RegisterDeviceRequest registerDeviceRequest = 1;
  optional RegisterDeviceResponse registerDeviceResponse = 2;
  optional StartSelfTestRequest startSelfTestRequest = 3;
  optional StartSelfTestResponse startSelfTestResponse = 4;
  optional StopSelfTestRequest stopSelfTestRequest = 5;
  optional StopSelfTestResponse stopSelfTestResponse = 6;
  optional UpdateFirmwareRequest updateFirmwareRequest = 7;
  optional UpdateFirmwareResponse updateFirmwareResponse = 8;
  optional SetLightRequest setLightRequest = 9;
  optional SetLightResponse setLightResponse = 10;
  optional GetStatusRequest getStatusRequest = 11;
  optional GetStatusResponse getStatusResponse = 12;
  optional ResumeScheduleRequest resumeScheduleRequest = 13;
  optional ResumeScheduleResponse resumeScheduleResponse = 14;
  optional SetEventNotificationsRequest setEventNotificationsRequest = 15;
  optional SetEventNotificationsResponse setEventNotificationsResponse = 16;
  optional GetFirmwareVersionRequest getFirmwareVersionRequest = 19;
  optional GetFirmwareVersionResponse getFirmwareVersionResponse = 20;
  optional SetScheduleRequest setScheduleRequest = 21;
  optional SetScheduleResponse setScheduleResponse = 22;
  optional SetConfigurationRequest setConfigurationRequest = 25;
  optional SetConfigurationResponse setConfigurationResponse = 26;
  optional GetPowerUsageHistoryRequest getPowerUsageHistoryRequest = 27;
  optional GetPowerUsageHistoryResponse getPowerUsageHistoryResponse = 28;
  optional GetActualPowerUsageRequest getActualPowerUsageRequest = 29;
  optional GetActualPowerUsageResponse getActualPowerUsageResponse = 30;
  optional SetRebootRequest setRebootRequest = 31;
  optional SetRebootResponse setRebootResponse = 32;
  optional SetTransitionRequest setTransitionRequest = 33;
  optional SetTransitionResponse setTransitionResponse = 34;
  optional GetConfigurationRequest getConfigurationRequest = 35;
  optional GetConfigurationResponse getConfigurationResponse = 36;
  optional ConfirmRegisterDeviceRequest confirmRegisterDeviceRequest = 37;
  optional ConfirmRegisterDeviceResponse confirmRegisterDeviceResponse = 38;
  optional UpdateDeviceSslCertificationRequest updateDeviceSslCertificationRequest = 39;
  optional UpdateDeviceSslCertificationResponse updateDeviceSslCertificationResponse = 40;
  optional SetDeviceVerificationKeyRequest setDeviceVerificationKeyRequest = 41;
  optional SetDeviceVerificationKeyResponse setDeviceVerificationKeyResponse = 42;
  optional SwitchFirmwareRequest switchFirmwareRequest = 43;
  optional SwitchFirmwareResponse switchFirmwareResponse = 44;
  optional SwitchConfigurationRequest switchConfigurationRequest = 45;
  optional SwitchConfigurationResponse switchConfigurationResponse = 46;
}

// ========= Device Installation
message RegisterDeviceRequest {
  required string deviceIdentification = 1; // [(nanopb).max_size = 41];
  required bytes ipAddress = 2; // [(nanopb).max_size = 4];
  required DeviceType deviceType = 3;
  required bool hasSchedule = 4;
  required uint32 randomDevice = 5; // 16 bits
}

message RegisterDeviceResponse {
  required Status status = 1;
  required string currentTime = 2; // [(nanopb).max_size = 15]; // - Format YYYYMMDDhhmmss
  required uint32 randomDevice = 3;
  required uint32 randomPlatform = 4;
}
```
optional LocationInfo locationInfo = 5; // Location information of device.

message StartSelfTestRequest {
  optional bool present = 1 [default = true];
}
message StartSelfTestResponse {
  required Status status = 1;
}
message StopSelfTestRequest {
  optional bool present = 1 [default = true];
  required Status status = 1;
  required bytes selfTestResult = 2; // [(nanopb).max_size = 1];
}

// ========= Firmware Management
message GetFirmwareVersionRequest {
  optional bool present = 1 [default = true];
}
message GetFirmwareVersionResponse {
  required string firmwareVersion = 1; // [(nanopb).max_size = 7]; // RXX
}
message UpdateFirmwareRequest {
  required string firmwareDomain = 1; // [(nanopb).max_size = 180]; // Server-name without
  required string firmwareUrl = 2; // [(nanopb).max_size = 255]; // Relative URL like this
}
message UpdateFirmwareResponse {
  required Status status = 1;
}
message SwitchFirmwareRequest {
  required string newFirmwareVersion = 1; // [(nanopb).max_size = 6]; // The version of th
}
message SwitchFirmwareResponse {
  required Status status = 1; // FIRMWARE_EVENTS_ACTIVATING Event will be sent, after the
}

// ========= Ad-Hoc & Status
message SetLightRequest {
  repeated LightValue values = 1; // [(nanopb).max_count = 6];
}
message SetLightResponse {
  required Status status = 1;
}
message GetStatusRequest {
  optional bool present = 1 [default = true];
}
message GetStatusResponse {
  required Status status = 1;
  repeated LightValue value = 2; // [(nanopb).max_count = 6];
  required LinkType preferredLinktype = 3;
  required LinkType actualLinktype = 4;
  required LightType lightType = 5;
  required uint32 eventNotificationMask = 6; // Bitmask for max 32 events, using N
  required uint32 numberOfOutputs = 7; // Hardware - The number of outputs o
  optional uint32 dcOutputVoltageMaximum = 8; // Hardware - DC output voltage MAXim
  optional uint32 dcOutputVoltageCurrent = 9; // Hardware - DC output current volt
  optional uint32 maximumOutputPowerOnDcOutput = 10; // Hardware - Maximum output power o
  optional bytes serialNumber = 11; // [(nanopb).max_size = 18]; // Hardware - Serial numb
  optional bytes macAddress = 12; // [(nanopb).max_size = 6]; // Hardware - MAC-address of
optional string hardwareId = 13;  // [(nanopb).min_size = 10, (nanopd).max_size = 25]; // Hardware - The internal flash memory.
optional uint32 internalFlashMemSize = 14; // Hardware - The internal flash memory.
optional uint32 externalFlashMemSize = 15; // Hardware - The external flash memory.
optional uint32 lastInternalTestResultCode = 16; // Hardware - The last internal test result.
optional uint32 startupCounter = 17; // Hardware - The startup counter.
optional string bootLoaderVersion = 18; // Software - The boot loader version.
optional string firmwareVersion = 19; // Software - The firmware version.
optional uint32 currentConfigurationBackUsed = 20; // [(nanopb).max_size = 6]; // Software - The current configuration bank in use.
optional string name = 21; // Device - The name of this device.
optional string currentTime = 22; // Device - Not UTC, the time used in RT_NTP SET is NOT supported!
optional string currentIp = 23; // Device - The current IP address of device.
}

message ResumeScheduleRequest {
  optional bytes index = 1; // [(nanopb).max_size = 1]; // Index number of connected light. required bool immediate = 2; // [default = true]; // Resume at next schedule item or
}
message ResumeScheduleResponse {
  required Status status = 1;
}

message SetRebootRequest {
  optional bool present = 1 [default = true];
}
message SetRebootResponse {
  required Status status = 1;
}

message SetTransitionRequest {
  required TransitionType transitionType = 1; // Night-Day or Day-Night transition.
  optional string time = 2; // [(nanopb).max_size = 7]; // Format hhmmss UTC.
}
message SetTransitionResponse {
  required Status status = 1;
}

message SetEventNotificationsRequest {
  required uint32 NotificationMask = 1; // Bitmask for max 32 events, using NotificationBit.
}
message SetEventNotificationsResponse {
  required Status status = 1;
}

message EventNotificationRequest {
  repeated EventNotification notifications = 1; // [(nanopb).max_count = 6];
}
message EventNotificationResponse {
  required Status status = 1;
}

message SetScheduleRequest {
  repeated Schedule schedules = 1; // [(nanopb).max_count = 50];
  optional PageInfo pageInfo = 2;
  required RelayType scheduleType = 3; // RT_NOT_SET is NOT supported!
}
message SetScheduleResponse {
  required Status status = 1;
}

message SetConfigurationRequest {
  optional LightType lightType = 1;
  optional DaliConfiguration daliConfiguration = 2; // (nanopb).min_size = 10, (nanopd).max_size = 25]; //
  optional RelayConfiguration relayConfiguration = 3; //
  optional uint32 shortTermHistoryIntervalMinutes = 4; //
}
optional LinkType preferredLinkType = 5;
optional MeterType meterType = 6;  // D
optional uint32 longTermHistoryInterval = 7;
optional LongTermHistoryIntervalType longTermHistoryIntervalType = 8;  // |
optional uint32 timeSyncFrequency = 9 [default = 86400];  // |
optional bytes deviceFixIpValue = 10;  // [(nanopb).max_count = 4];  // T
optional bytes netMask = 11;  // [(nanopb).max_count = 4];  // N
optional bytes gateWay = 12;  // [(nanopb).max_count = 4];  // G
optional bool isDhcpEnabled = 13 [default = true];  // I
  optional bool isTlsEnabled = 14;  // I
  optional uint32 oslpBindPortNumber = 15;  // I
  optional string commonNameString = 16 [default = 'TLS Test'];  // [(default = 'TLS Test'),
    optional uint32 communicationTimeout = 14 [default = 20];  // |
  optional uint32 communicationNumberOfRetries = 15 [default = 3];  // |
  optional uint32 communicationNumberOfConnectionTrials = 16 [default = 60];  // |
  optional bytes osdpIpAddress = 17;  // [(nanopb).max_count = 4];  // T
  optional bytes deviceFixIpValue = 18;  // [(nanopb).max_count = 4];  // T
optional bool isTestButtonEnabled = 19 [default = true];  // |
optional bool isAutomaticSummerTimingEnabled = 20 [default = true];  // |
optional sint32 astroGateSunRiseOffset = 21 [default = 0];  // |
optional sint32 astroGateSunSetOffset = 22 [default = 0];  // |
repeated uint32 switchingDelay = 23;  // [(nanopb).max_count = 4];  // S
repeated RelayMatrix relayLinking = 24;  // F
optional bool relayRefreshing = 25 [default = true];  // |
optional string summerTimeDetails = 26 [default = '0360100'];  // [(default = '0360100'),(na
optional string winterTimeDetails = 27 [default = '1060200'];  // [(default = '1060200'),(na
  // winterTimeDetails string, winterTimeDetails:
  // MMMWmm
  // where: (note, north hemisphere summer begins at the end of march)
  // MMM: month
  // M: day of the week (1- Monday, 6- Sunday)
  // HH: hour of the changing time
  // mm: minutes of the changing time
message SetConfigurationResponse {
  required Status status = 1;
}

message GetConfigurationRequest {
  optional bool present = 1 [default = true];
}

message GetConfigurationResponse {
  required Status status = 1;
  optional LightType lightType = 2;
  optional DalConfiguration dalConfiguration = 3;  // |
  optional RelayConfiguration relayConfiguration = 4;  // |
  optional uint32 shortTermHistoryIntervalMinutes = 5;  // I
  optional LinkType preferredLinkType = 6;  // I
  optional MeterType meterType = 7;  // D
  optional uint32 longTermHistoryInterval = 8;  // |
  optional LongTermHistoryIntervalType longTermHistoryIntervalType = 9;  // |
  optional uint32 timeSyncFrequency = 10 [default = 86400];  // T
  optional bytes netMask = 11;  // [(nanopb).max_count = 4];  // N
  optional bytes gateWay = 12;  // [(nanopb).max_count = 4];  // G
  optional bool isDhcpEnabled = 14 [default = true];  // I
  optional bool isTlsEnabled = 15;  // I
  optional uint32 oslpBindPortNumber = 16;  // I
  optional string commonNameString = 17 [default = 'TLS Test'];  // [(default = 'TLS Test'),
    optional uint32 communicationTimeout = 15 [default = 20];  // C
    optional uint32 communicationNumberOfRetries = 16 [default = 3];  // C
    optional uint32 communicationNumberOfConnectionTrials = 17 [default = 60];  // C
    optional bytes osdpIpAddress = 18;  // [(nanopb).max_count = 4];  // T
    optional uint32 oslpBindPortNumber = 19;  // T
    optional bool isTestButtonEnabled = 20 [default = true];  // |
    optional bool isAutomaticSummerTimingEnabled = 21 [default = true];  // |
    optional sint32 astroGateSunRiseOffset = 22 [default = 0];  // |
    optional sint32 astroGateSunSetOffset = 23 [default = 0];  // |
    repeated uint32 switchingDelay = 24;  // [(nanopb).max_count = 4];  // S
repeated RelayMatrix relayLinking = 25;  // Relay linking is a software linking, to may link each relay with each other relay. It is a matrix. Example, if relay 1 is linked with relay 3, if relay 1 will be switched (by
optional bool relayRefreshing = 26 [default = true];  //
optional string summerTimeDetails = 27 [default = '0360100']; // [default = '0360100'], (nanopb).max_count = 7
optional string winterTimeDetails = 28 [default = '1060200']; // [default = '1060200'], (nanopb).max_count = 7
}
message SwitchConfigurationRequest {
  required bytes newConfigurationSet = 1; // [nanopb].max_count = 1]; // The index of the
}
message SwitchConfigurationResponse {
  required Status status = 1; // FIRMWARE_EVENTS_CONFIGURATION_CHANGED Event will be sent,
}
message ConfirmRegisterDeviceRequest {
  required uint32 randomDevice = 1;
  required uint32 randomPlatform = 2;
}
message ConfirmRegisterDeviceResponse {
  required Status status = 1;
  required uint32 randomDevice = 2;
  required uint32 randomPlatform = 3;
  required uint32 sequenceWindow = 4;
}

======== Monitoring
// Deprecated, no longer supported by the platform.
message GetPowerUsageHistoryRequest {
  required TimePeriod timePeriod = 1;
  optional uint32 page = 2;
  required HistoryTermType termType = 3;
}
// Deprecated, no longer supported by the platform.
message GetPowerUsageHistoryResponse {
  required Status status = 1;
  repeated PowerUsageData powerUsageData = 2; // [nanopb].max_count = 20];
  optional PageInfo pageInfo = 3;
}
// Deprecated, no longer supported by the platform.
message GetActualPowerUsageRequest {
  optional bool present = 1 [default = true];
}
// Deprecated, no longer supported by the platform.
message GetActualPowerUsageResponse {
  required Status status = 1;
  required PowerUsageData powerUsageData = 2;
}

======== Certificate Management
message UpdateDeviceSslCertificationRequest {
  required string certificateDomain = 1; // [nanopb].max_size = 100]; // The domain name
  required string certificateUrl = 2; // [nanopb].max_size = 255]; // The relative path
}
message UpdateDeviceSslCertificationResponse {
  required Status status = 1;
}

======== Key Management
message SetDeviceVerificationKeyRequest {
  required bytes certificateChunk = 1; // [nanopb].max_size = 138]; // Verification key
}
message SetDeviceVerificationKeyResponse {
  required Status status = 1;
}
/ ===== Type

message LocationInfo {
  optional sint32 timeOffset = 1; // Correction in minutes with respect to UTC.
  optional sint32 latitude = 2; // Divide by 1000000 to get float value.
  optional sint32 longitude = 3; // Divide by 1000000 to get float value.
}

message LightValue {
  optional bytes index = 1; // [(nanopb).max_size = 1]; // Index number of connected light
  required bool on = 2;
  optional bytes dimValue = 3; // [(nanopb).max_size = 1]; // 1 - 100 %
}

message EventNotification {
  required Event event = 1;
  optional bytes index = 2; // [(nanopb).max_size=1];
  optional string description = 3; // [(nanopb).max_size = 81];
  optional string timestamp = 4; // [(nanopb).max_size = 15]; // - Format YYYYMMDDhhmmss UTC,
}

message Schedule {
  required Weekday weekday = 1;
  optional string startDay = 2; // [(nanopb).max_size = 9]; // - Format YYYYMMDD UTC, indic
  optional string endDay = 3; // [(nanopb).max_size = 9]; // - Format YYYYMMDD UTC, includ
  required ActionType actionTime = 4;
  optional string time = 5; // [(nanopb).max_size = 7]; // - Format hhmmss localtime set w
  optional Window window = 6; // Window to wait for light sensor trigger.
  repeated LightValue value = 7; // [(nanopb).max_count = 6];
  optional TriggerType triggerType = 8; // React to setTransition or switch astronomical.
  optional uint32 minimumLightsOn = 9; // Minimal time (in seconds) the lights should burn.
  optional uint32 index = 10; // Index of schedule entry in the schedule list.
  optional bool isEnabled = 11; // Is this schedule entry enabled?
}

message Window {
  required uint32 minutesBefore = 1; // Minutes before sunset / sunrise.
  required uint32 minutesAfter = 2; // Minutes after sunset / sunrise.
}

message DaliConfiguration {
  optional bytes numberOfLights = 1; // [(nanopb).max_size = 1]; // Number of lights connec
  repeated IndexAddressMap addressMap = 2; // [(nanopb).max_count = 4];
}

message RelayConfiguration {
  repeated IndexAddressMap addressMap = 1; // [(nanopb).max_count = 6];
}

message RelayMatrix {
  required bytes masterRelayIndex = 1; // [(nanopb).max_count = 1];
  required bool masterRelayOn = 2; // [(nanopb).max_count = 1];
  optional bytes indicesOfControlledRelaysOn = 3; // [(nanopb).max_count = 4]; // IndexNu
  optional bytes indicesOfControlledRelaysOff = 4; // [(nanopb).max_count = 4]; // IndexNu
}

message IndexAddressMap {
  required bytes index = 1; // [(nanopb).max_size = 1]; // External index, for example 1.
  required bytes address = 2; // [(nanopb).max_size = 1]; // Internal address, for example
  required RelayType relayType = 3;
}

message PageInfo {
  required uint32 currentPage = 1; // Pages start from 1.
  required uint32 pageSize = 2;
  required uint32 totalPages = 3;
}

// Deprecated, no longer supported by the platform.
message TimePeriod {
  required string startTime = 1; // [(nanopb).max_size = 15]; // - Format YYYYMMDDhhmmss UTC
  required string endTime = 2; // [(nanopb).max_size = 15]; // - format YYYYMMDDhhmmss UTC
}
// Deprecated, no longer supported by the platform.
message PowerUsageData {
  required string recordTime = 1; // [(nanopb).max_size = 15]; // Record time - format
  required MeterType meterType = 2; // Meter type (P1, Pulse,
  required uint64 totalConsumedEnergy = 3; // Electricity delivered
  required uint32 actualConsumedPower = 4; // Actual Electricity pow
  optional PsldData psldData = 5;
  optional SsldData ssldData = 6;
}

// Deprecated, no longer supported by the platform.
message PsldData {
  required uint32 totalLightingHours = 1; // Total lighting hours
}

// Deprecated, no longer supported by the platform.
message SsldData {
  required uint32 totalLightingMinutes = 2; // Total lighting minutes for lighting relay
}

// ========== Enumerations
// ========== Event Notification

enum NotificationBit {
  DIAG_EVENTS = 1;
  HARDWARE_FAILURE = 2;
  LIGHT_EVENTS = 4; // For example LightValue changes.
  TARIFF_EVENTS = 8; // For example Tariff changes.
  MONITOR_EVENTS = 16; // For example monitor buffer is almost full.
  FIRMWARE_EVENTS = 32; // For example firmware activation.
  COMM_EVENTS = 64; // For example alternative channel.
  SECURITY_EVENTS = 128; // For example out of sequence.
}

// Events must map to their notification bit:
// EG: 0000-0999 =1
// 1000-1999 =2
// 2000-2999 =4
// 3000-3999 =8
// 4000-4999 =16
// 5000-5999 =32
// 6000-6999 =64
// 7000-7999 =128
// OR to check 2^((event num)/1000)=notification bit

enum Event {
  // 0 - 999 Diagnostics
  DIAG_EVENTS_GENERAL = 0; // Multi-purpose event, see description of event n
  DIAG_EVENTS_UNKNOWN_MESSAGE_TYPE = 1; // Message type unknown by device.

  // 1000 - 1999 Hardware Failures
  HARDWARE_FAILURE_RELAY = 1000; // Index indicates relay (not supported ye
  HARDWARE_FAILURE_FLASH_WRITE_ERROR = 1001; // Error while writing to flash memory.
  HARDWARE_FAILURE_FLASH_MEMORY_CORRUPT = 1002; // Error while reading from flash memory
  HARDWARE_FAILURE_RTC_NOT_SET = 1003; // Real Time Clock has not set.

  // 2000 - 2999 Light Events

LIGHT_EVENTS_LIGHT_ON = 2000; // Index indicates light.
LIGHT_EVENTS_LIGHT_OFF = 2001; // Index indicates light.
LIGHT_FAILURE_DALI_COMMUNICATION = 2500; // DALI communication failure.
LIGHT_FAILURE_BALLAST = 2501; // Ballast failure detected (DALI only).
LIGHT_FAILURE_TARIFF_SWITCH_ATTEMPT = 2502; // Attempt to switch an end-point configured

// 3000 - 3999 Tariff Events
TARIFF_EVENTS_TARIFF_ON = 3000; // Tariff switched on.
TARIFF_EVENTS_TARIFF_OFF = 3001; // Tariff switched off.

// 4000 - 4999
MONITOR_EVENTS_LONG_BUFFER_FULL = 4000; // Long term monitoring buffer overrun occurs.
MONITOR_FAILURE_P1_COMMUNICATION = 4500; // P1 meter could not be read.
MONITOR_SHORT_DETECTED = 4600; // A short has been detected.
MONITOR_SHORT_RESOLVED = 4601; // A short has been resolved.
MONITOR_DOOR_OPENED = 4700; // Indicates that the enclosure of the device has been opened.
MONITOR_DOOR_CLOSED = 4701; // Indicates that the enclosure of the device has been closed.
MONITOR_EVENTS_TEST_RELAY_ON = 4702; // Relay was switched on by self-test function.
MONITOR_EVENTS_TEST_RELAY_OFF = 4703; // Relay was switched off by self-test function.
MONITOR_EVENTS_SHORT_DETECTED = 4600; // A short has been detected.
MONITOR_EVENTS_SHORT_RESOLVED = 4601; // A short has been resolved.
MONITOR_EVENTS_LOCAL_MODE = 4900; // Device switched to local mode.
MONITOR_EVENTSREMOTE_MODE = 4901; // Device switched to remote mode.

// 5000 - 5999 Firmware Events
FIRMWARE_EVENTS_ACTIVATING = 5000; // Start activating new firmware, after download.
FIRMWARE_EVENTSDOWNLOAD_NOTFOUND = 5501; // Download of firmware failed, i.e. location incorrect.
FIRMWARE_EVENTSDOWNLOAD_FAILED = 5502; // Download of firmware failed, image incorrect.
FIRMWARE_EVENTS_CONFIGURATION_CHANGED = 5503; // Configuration changed from one bank to another.

// 6000 - 6999
COMM_EVENTS_ALTERNATIVE_CHANNEL = 6000; // Alternative channel selected for communication.
COMM_EVENTS_RECOVERED_CHANNEL = 6001; // Communication has been recovered for this channel.

// 7000 - 7999
SECURITY_EVENTS_OUT_OF_SEQUENCE = 7000; // Out of sequence occurred and sequence number renegotiated.
SECURITY_EVENTS_OSLP_VERIFICATION_FAILED = 7001; // OSLP message could not be verified.
SECURITY_EVENTS_INVALID_CERTIFICATE = 7002; // Invalid TLS certificate.

// ======== Enums
enum TriggerType {
    TT_NOT_SET = 0;
    LIGHT_TRIGGER = 1;
    ASTRONOMICAL = 2;
}

eenum TransitionType {
    NIGHT_DAY = 0;
    DAY_NIGHT = 1;
}

eenum Weekday {
    MONDAY = 1;
    TUESDAY = 2;
    WEDNESDAY = 3;
    THURSDAY = 4;
    FRIDAY = 5;
    SATURDAY = 6;
    SUNDAY = 7;
    WEEKDAY = 8;
    WEEKEND = 9;
    ABSOLUTEDAY = 10;
    ALL = 11;
}

eenum ActionType {
    ABSOLUTETIME = 1;
    SUNRISE = 2;
    SUNSET = 3;
}

eenum DeviceType {
PSLD = 0;
SSLD = 1;
}

enum Status {
    OK = 0;
    FAILURE = 1; // General failure.
    REJECTED = 2; // Request received in wrong state.
}

enum LightType {
    LT_NOT_SET = 0;
    RELAY = 1;
    ONE_TO_TEN_VOLT = 2;
    ONE_TO_TEN_VOLT_REVERSE = 3;
    DALI = 4;
}

enum RelayType {
    RT_NOT_SET = 0;
    LIGHT = 1;
    TARIFF = 2;
}

// Deprecated, no longer supported by the platform.
enum MeterType {
    MT_NOT_SET = 0;
    P1 = 1;
    PULSE = 2;
    AUX = 3;
}

enum LinkType {
    LINK_NOT_SET = 0;
    GPRS = 1;
    CDMA = 2;
    ETHERNET = 3;
}

// Deprecated, no longer supported by the platform.
enum LongTermIntervalType {
    LT_INT_NOT_SET = 0;
    DAYS = 1;
    MONTHS = 2;
}

// Deprecated, no longer supported by the platform.
enum HistoryTermType {
    Short = 0;
    Long = 1;
}
RegisterDevice

RegisterDevice messages

Description

The device registration is a 2 step process. First RegisterDeviceRequest and RegisterDeviceResponse are exchanged between device and platform. Second ConfirmRegisterDeviceRequest and ConfirmRegisterDeviceResponse messages are exchanged.

Request that notifies the platform a device which wants to register. During the registration the sequence number is reset to a random value the platform is notified if the device has a light schedule, the type of the device, the device identification, and the device communicates its IP address to the platform. Also a random number is determined by the device and this 'randomDevice' should be present in the response form the platform.

Response which holds the time of the platform so the device can synchronize the time, contains location information for the device like GPS coordinates and Daylight Saving Time information. The device will sent ConfirmRegisterDeviceRequest after receiving the RegisterDeviceResponse. Also a random number is determined by the platform and this 'randomPlatform' should be present in the next request 'ConfirmRegisterDeviceRequest' by the device.

Message definitions

```protobuf
message RegisterDeviceRequest {
  required string deviceIdentification = 1; // [(nanopb).max_size = 41];
  required bytes ipAddress = 2; // [(nanopb).max_size = 4];
  required DeviceType deviceType = 3;
  required bool hasSchedule = 4;
  required uint32 randomDevice = 5; // 16 bits
}

message RegisterDeviceResponse {
  required Status status = 1;
  required string currentTime = 2; // [(nanopb).max_size = 15];// - format YYYYMMDDhhmmss
  required uint32 randomDevice = 3;
  required uint32 randomPlatform = 4;
  optional LocationInfo locationInfo = 5; // Location information of device
}
```

Datatypes

```protobuf
enum DeviceType {
  PSLD = 0;
  SSLD = 1;
}

enum Status {
  OK = 0;
  FAILURE = 1; // general failure
  REJECTED = 2; // request received in wrong state
}
```

```protobuf
message LocationInfo {
  optional sint32 timeOffset = 1; // correction in minutes with respect to UTC
  optional sint32 latitude = 2; // divide by 1000000 to get float value
  optional sint32 longitude = 3; // divide by 1000000 to get float value
}
```

Example

**OSLP** RegisterDeviceRequest sent from 'device-01' to platform:

```protobuf
registerDeviceRequest {
  deviceIdentification: "device-01"
  ipAddress: "#\000\000\001"
  deviceType: SSLD
  hasSchedule: false
  randomDevice: 13246
}
```

**OSLP** RegisterDeviceResponse sent from platform to 'device-01':

```protobuf
```
registerDeviceResponse {
  status: OK
  currentTime: "20160106135210"
  randomDevice: 13246
  randomPlatform: 44765
  locationInfo {
    timeOffset: 60
    latitude: 50889228
    longitude: 5974140
  }
}
ConfirmRegisterDevice

ConfirmRegisterDevice messages

Description
Request which contains the 2 random numbers from RegisterDeviceRequest and RegisterDeviceResponse. The numbers should match with the previous request and response and this is checked by the platform.

Response which contains the sequenceWindow which is the maximum allowed difference between sequence numbers for future messages. Further the response contains the 2 random numbers from the ConfirmRegisterDeviceRequest. The numbers should match with the previous request and response and this is checked by the device.

Message definitions

message ConfirmRegisterDeviceRequest {
  required uint32 randomDevice = 1;
  required uint32 randomPlatform = 2;
}

message ConfirmRegisterDeviceResponse {
  required Status status = 1;
  required uint32 randomDevice = 2;
  required uint32 randomPlatform = 3;
  required uint32 sequenceWindow = 4;
}

Datatypes
enum Status {
  OK = 0;
  FAILURE = 1; // general failure
  REJECTED = 2; // request received in wrong state
}

Example
OSLP ConfirmRegisterDeviceRequest sent from ‘device-01’ to platform:

confirmRegisterDeviceRequest {
  randomDevice: 13246
  randomPlatform: 44765
}

OSLP ConfirmRegisterDeviceResponse sent from platform to ‘device-01’:

confirmRegisterDeviceResponse {
  status: OK
  randomDevice: 13246
  randomPlatform: 44765
  sequenceWindow: 6
}
GetConfiguration

GetConfiguration messages

Description

Request to fetch the current configuration of a device.

Response communicates if the request was executed. If 'status = OK' then the optional fields will be partly populated. Note that DaliConfiguration is only present for devices with 'lightType = DALI', which are of device type PSLD. Note that RelayConfiguration is only present for devices with 'lightType = RELAY | ONE_TO_TEN_VOLT | ONE_TO_TEN_VOLT_REVERSE', which are of device type SSLD.

Message definitions

message GetConfigurationRequest {
  optional bool present = 1 [default = true];
}

message GetConfigurationResponse {
  required Status status = 1;
  optional LightType lightType = 2;
  optional DaliConfiguration daliConfiguration = 3; //
  optional RelayConfiguration relayConfiguration = 4; //
  optional uint32 shortTermHistoryIntervalMinutes = 5;
  optional LinkType preferredLinkType = 7;
  optional uint32 longTermHistoryInterval = 8;
  optional LongTermIntervalType longTermHistoryIntervalType = 9;
  optional uint32 timeSyncFrequency = 10 [default = 86400]; //
  optional bytes deviceFixIpValue = 11; //[(nanopb).max_count = 4]; //T
  optional bytes netMask = 12; //[(nanopb).max_count = 4]; //H
  optional bytes gateWay = 13; //[(nanopb).max_count = 4]; //G
  optional bool relayRefreshingEnabled = 14 [default = true]; //I
  optional bool isTlsEnabled = 15; //D
  optional uint32 oslpBindPortNumber = 16; //I
  optional string commonNameString = 17 [default = 'TLS Test']; //[default = 'TLS Test', (nanopb).max_count = 1];
  optional uint32 communicationTimeout = 18 [default = 20]; //C
  optional uint32 communicationNumberOfRetries = 19 [default = 3]; //
  optional uint32 communicationPauseTimeBetweenConnectionTrials = 20 [default = 60]; //
  optional bytes osgpIpAddress = 21; //[(nanopb).max_count = 4]; //I
  optional uint32 osgpPortNumber = 22; //I
  optional bool isTestButtonEnabled = 23 [default = true]; //I
  optional bool isAutomaticSummerTimingEnabled = 24 [default = true]; //I
  optional sint32 astroGateSunRiseOffset = 25 [default = 0]; //'
  optional sint32 astroGateSunSetOffset = 26 [default = 0]; //'
  repeated uint32 switchingDelay = 27; //[(nanopb).max_count = 4]; //S
  repeated RelayMatrix relayLinking = 28; //R
  optional bool relayRefreshing = 29 [default = true]; //I
  optional string summerTimeDetails = 30 [default = '0360100']; //[default = '0360100', (nanopb).max_count = 1];
  optional string winterTimeDetails = 31 [default = '1060200']; //[default = '1060200', (nanopb).max_count = 1];
}

Datatypes

enum Status {
  OK = 0;
  FAILURE = 1; // general failure
  REJECTED = 2; // request received in wrong state
}

enum LightType {
  LT_NOT_SET = 0;
  RELAY = 1;
  ONE_TO_TEN_VOLT = 2;
  ONE_TO_TEN_VOLT_REVERSE = 3;
  DALI = 4;
}

message DaliConfiguration {
  optional bytes numberOfLights = 1; //[(nanopb).max_size = 1]; // number of lights connected

repeated IndexAddressMap addressMap = 2; // [(nanopb).max_count = 4];
}

message RelayConfiguration {
  repeated IndexAddressMap addressMap = 1; // [(nanopb).max_count = 6];
}

message IndexAddressMap {
  required bytes index = 1;  // [(nanopb).max_size = 1]; // external index, for example 1
  required bytes address = 2; // [(nanopb).max_size = 1]; // internal address, for example 2
  required RelayType relayType = 3;
}

enum RelayType {
  RT_NOT_SET = 0;
  LIGHT = 1;
  TARIFF = 2;
}

enum LinkType {
  LINK_NOT_SET = 0;
  GPRS = 1;
  CDMA = 2;
  ETHERNET = 3;
}

enum MeterType {
  MT_NOT_SET = 0;
  P1 = 1;
  PULSE = 2;
  AUX = 3;
}

enum LongTermIntervalType {
  LT_INT_NOT_SET = 0;
  DAYS = 1;
  MONTHS = 2;
}

message RelayMatrix {
  required bytes masterRelayIndex = 1; // [(nanopb).max_count = 1];
  required bool masterRelayOn = 2; // [(nanopb).max_count = 1];
  optional bytes indicesOfControlledRelaysOn = 3; // [(nanopb).max_count = 4]; // IndexNumber of output Relay to switch ON if Master Relay state changes as determined by masterRelayOn.
  optional bytes indicesOfControlledRelaysOff = 4; // [(nanopb).max_count = 4]; // IndexNumber of output Relay to switch OFF if Master Relay state changes as determined by MasterRelayOff.
}

Example

Soap requests and responses sent to and from platform:

```xml
  xmlns:ns2="http://www.opensmartgridplatform.org/sci"
  xmlns:ns3="http://www.opensmartgridplatform.org/sci"
  xmlns:ns4="http://www.opensmartgridplatform.org/sci"
  xmlns:ns5="http://www.opensmartgridplatform.org/sci">
  <soapenv:Header>
    <com:OrganisationIdentification>LianderNetManagement</com:OrganisationIdentification>
    <com:UserName>Kevin</com:UserName>
    <com:ApplicationName>SoapUI</com:ApplicationName>
  </soapenv:Header>
  <soapenv:Body>
    <con:GetConfigurationRequest>
      <con:DeviceIdentification>device-01</con:DeviceIdentification>
    </con:GetConfigurationRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

```xml
  xmlns:ns3="http://www.opensmartgridplatform.org/sci"
  xmlns:ns4="http://www.opensmartgridplatform.org/sci"
  xmlns:ns5="http://www.opensmartgridplatform.org/sci">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:GetConfigurationAsyncResponse xmlns:ns3="http://www.opensmartgridplatform.org/sci"
      xmlns:ns4="http://www.opensmartgridplatform.org/sci"
      xmlns:ns5="http://www.opensmartgridplatform.org/sci">
      <ns3:CorrelationUid>LianderNetManagement|||device-01|||201607142028655</ns3:CorrelationUid>
      <ns3:DeviceId>device-01</ns3:DeviceId>
      <ns2:AsyncResponse/>
    </ns2:GetConfigurationAsyncResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
<ns2:RelayLinking>
  <ns2:MasterRelayIndex>2</ns2:MasterRelayIndex>
  <ns2:MasterRelayOn>false</ns2:MasterRelayOn>
  <ns2:IndicesOfControlledRelaysOn>4</ns2:IndicesOfControlledRelaysOn>
  <ns2:IndicesOfControlledRelaysOff>4</ns2:IndicesOfControlledRelaysOff>
</ns2:RelayLinking>

<ns2:RelayRefreshing>false</ns2:RelayRefreshing>
<ns2:SummerTimeDetails>2016-03-27T01:00:00.000+01:00</ns2:SummerTimeDetails>
<ns2:WinterTimeDetails>2016-10-30T02:00:00.000+02:00</ns2:WinterTimeDetails>
</ns2:Configuration>
</ns2:GetConfigurationResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

OSLP GetConfigurationRequest message sent to 'device-01':
getConfigurationRequest {
}

OSLP GetConfigurationResponse message sent to platform:
getConfigurationResponse {
  status: OK
  lightType: RELAY
  relayConfiguration {
    addressMap {
      index: "\001"
      address: "\001"
      relayType: TARIFF
    }
    addressMap {
      index: "\002"
      address: "\002"
      relayType: LIGHT
    }
    addressMap {
      index: "\003"
      address: "\003"
      relayType: LIGHT
    }
    addressMap {
      index: "\004"
      address: "\004"
      relayType: LIGHT
    }
  }
  shortTermHistoryIntervalMinutes: 15
  preferredLinkType: ETHERNET
  meterType: MT_NOT_SET
  longTermHistoryInterval: 1
  longTermHistoryIntervalType: LT_INT_NOT_SET
  timeSyncFrequency: 86400
  deviceFixIpValue: "\300\250\000d"
  netMask: "\377\377\377\000"
  gateWay: "\300\250\000\001"
  isDhcpEnabled: false
  isTlsEnabled: true
  oslbpBindPortNumber: 1234
  commonNameString: "TLS Test"
  communicationTimeout: 30
  communicationNumberOfRetries: 5
  communicationPauseTimeBetweenConnectionTrials: 120
  osgpIpAddress: "\250?aA"
  osgpPortNumber: 12122
  isTestButtonEnabled: false
  isAutomaticSummerTimingEnabled: false
  astroGateSunRiseOffset: -15
  astroGateSunSetOffset: 15
  switchingDelay: 1
  switchingDelay: 2
  switchingDelay: 3
switchingDelay: 4
relayLinking {
    masterRelayIndex: "\002"
    masterRelayOn: false
    indicesOfControlledRelaysOn: "\003\004"
    indicesOfControlledRelaysOff: "\003\004"
}
relayRefreshing: false
summerTimeDetails: "0366100"
winterTimeDetails: "1060200"
SetConfiguration

SetConfiguration messages

Description

Request to push configuration settings to a device.

Response communicates status.

Message definitions

message SetConfigurationRequest {
  optional LightType lightType = 1;
  optional DalIConfiguration daliConfiguration = 2;
  optional RelayConfiguration relayConfiguration = 3;
  optional uint32 shortTermHistoryIntervalMinutes = 4;
  optional LinkType preferredLinkType = 5;
  optional MeterType meterType = 6;
  optional uint32 longTermHistoryInterval = 7;
  optional LongTermHistoryIntervalType longTermHistoryIntervalType = 8;
  optional uint32 timeSyncFrequency = 9 [default = 86400];
  optional bytes deviceFixIpValue = 10; // [(nanopb).max_count = 4];
  optional bytes netMask = 11; // [(nanopb).max_count = 4];
  optional bytes gateWay = 12; // [(nanopb).max_count = 4];
  optional bool isDhcpEnabled = 13 [default = true];
  optional bool isTlsEnabled = 14; // D
  optional uint32 oslpBindPortNumber = 15; // D
  optional string commonNameString = 16 [default = 'TLS Test']; //[(default = 'TLS Test'),
  optional uint32 communicationTimeout = 17 [default = 20]; // C
  optional uint32 communicationNumberOfRetries = 18 [default = 3]; // C
  optional uint32 communicationPauseTimeBetweenConnectionTrials = 19 [default = 60]; // C
  optional bytes osgpIpAddress = 20; // [(nanopb).max_count = 4];
  optional uint32 osgpPortNumber = 21; // G
  optional bool isTestButtonEnabled = 22 [default = true];
  optional bool isAutomaticSummerTimingEnabled = 23 [default = true]; // H
  optional uint32 astroGateSunRiseOffset = 24 [default = 0];
  optional uint32 astroGateSunSetOffset = 25 [default = 0];
  repeated uint32 switchingDelay = 26; // [(nanopb).max_count = 4];
  repeated RelayMatrix relayLinking = 27; // R
  optional bool relayRefreshing = 28 [default = true];
  optional string summerTimeDetails = 29 [default = '0360180']; //[(default = '0360180'),
  optional string winterTimeDetails = 30 [default = '1060280']; //[(default = '1060280'),
  repeated IndexAddressMap addressMap = 2; // [(nanopb).max_count = 4];
}

// summerTimeDetails string, winterTimeDetails:
// (MM)HHmi
// where: (note, north hemisphere summer begins at the end of march)
// MM: month
// HH: hour of the changing time
// mi: minutes of the changing time

message SetConfigurationResponse {
  required Status status = 1;
}

Datatypes

enum LightType {
  LT_NOT_SET = 0;
  RELAY = 1;
  ONE_TO_TEN_VOLT = 2;
  ONE_TO_TEN_VOLT_REVERSE = 3;
  DALI = 4;
}

message DalIConfiguration {
  optional bytes numberOfLights = 1; // [(nanopb).max_size = 1]; // number of lights connected
  repeated IndexAddressMap addressMap = 2; // [(nanopb).max_count = 4];
}
message RelayConfiguration {
  repeated IndexAddressMap addressMap = 1; // [(nanopb).max_count = 6];
}

message IndexAddressMap {
  required bytes index = 1; // [(nanopb).max_size = 1]; // external index, for example 1
  required bytes address = 2; // [(nanopb).max_size = 1]; // internal address, for example
  required RelayType relayType = 3;
}

enum RelayType {
  RT_NOT_SET = 0;
  LIGHT = 1;
  TARIFF = 2;
}

denum LinkType {
  LINK_NOT_SET = 0;
  GPRS = 1;
  CDMA = 2;
  ETHERNET = 3;
}

denum MeterType {
  MT_NOT_SET = 0;
  P1 = 1;
  PULSE = 2;
  AUX = 3;
}

denum LongTermIntervalType {
  LT_INT_NOT_SET = 0;
  DAYS = 1;
  MONTHS = 2;
}

message RelayMatrix {
  required bytes masterRelayIndex = 1; // [(nanopb).max_count = 1];
  required bool masterRelayOn = 2; // [(nanopb).max_count = 1];
  optional bytes indicesOfControlledRelaysOn = 3; // [(nanopb).max_count = 4]; // IndexNumber of output Relay to switch ON if Master Relay state changes as determined by masterRelayOn.
  optional bytes indicesOfControlledRelaysOff = 4; // [(nanopb).max_count = 4]; // IndexNumber of output Relay to switch OFF if Master Relay state changes as determined by MasterRelayOff.
}

denum Status {
  OK = 0;
  FAILURE = 1; // general failure
  REJECTED = 2; // request received in wrong state
}

Example
Soap requests and responses sent to and from platform:

Example
Soap requests and responses sent to and from platform:
<con:IndicesOfControlledRelaysOn>4</con:IndicesOfControlledRelaysOn>
<!-- List of RelayMatrixIndex, max 4 entries containing a single index between 1 and 4 -->
<con:IndicesOfControlledRelaysOff>1</con:IndicesOfControlledRelaysOff>
<con:IndicesOfControlledRelaysOff>2</con:IndicesOfControlledRelaysOff>
<con:IndicesOfControlledRelaysOff>3</con:IndicesOfControlledRelaysOff>
<con:IndicesOfControlledRelaysOff>4</con:IndicesOfControlledRelaysOff>
</con:RelayLinking>
<con:RelayLinking>
<!-- List of RelayMatrixIndex, max 4 entries containing a single index between 1 and 4 -->
<con:MasterRelayIndex>2</con:MasterRelayIndex>
<con:MasterRelayOn>true</con:MasterRelayOn>
</con:RelayLinking>
<con:RelayLinking>
<!-- List of RelayMatrixIndex, max 4 entries containing a single index between 1 and 4 -->
<con:MasterRelayIndex>3</con:MasterRelayIndex>
<con:MasterRelayOn>true</con:MasterRelayOn>
</con:RelayLinking>
<con:RelayRefreshing>true</con:RelayRefreshing>
<con:SummerTimeDetails>2016-03-27T01:00:00.000+01:00</con:SummerTimeDetails>
<con:WinterTimeDetails>2016-10-30T02:00:00.000+02:00</con:WinterTimeDetails>
</con:Configuration>
</soapenv:Body>
</soapenv:Envelope>

OSLP SetConfigurationRequest sent to 'device-01':

```
setConfigurationRequest {
  lightType: RELAY
  relayConfiguration {
    addressMap {
      index: "\001"
    }
  }
```
Grid eXchange Fabric Documentation

OSLP v0.6.1

address: "\001"
  relayType: TARIFF
}
addressMap {
  index: "\002"
  address: "\002"
  relayType: LIGHT
}
addressMap {
  index: "\003"
  address: "\003"
  relayType: LIGHT
}
addressMap {
  index: "\004"
  address: "\004"
  relayType: LIGHT
}
}
shortTermHistoryIntervalMinutes: 15
preferredLinkType: ETHERNET
meterType: PULSE
longTermHistoryInterval: 1
longTermHistoryIntervalType: DAYS
timeSyncFrequency: 864000
deviceFixIpValue: "\300\250\000n"
netMask: "\377\377\377\000"
gateWay: "\300\250\000\001"
isDhcpEnabled: false
isTlsEnabled: false
oslpBindPortNumber: 1234
commonNameString: "TLS Test"
communicationTimeout: 15
communicationNumberOfRetries: 2
communicationPauseTimeBetweenConnectionTrials: 120
osgpIpAddress: "\300\250\000d"
osgpPortNumber: 12122
isTestButtonEnabled: false
isAutomaticSummerTimingEnabled: false
astroGateSunRiseOffset: -15
astroGateSunSetOffset: 15
switchingDelay: 100
switchingDelay: 200
switchingDelay: 300
switchingDelay: 400
relayLinking {
  masterRelayIndex: "\001"
  masterRelayOn: true
  indicesOfControlledRelaysOn: "\001\002\003\004"
  indicesOfControlledRelaysOff: "\001\002\003\004"
}
relayLinking {
  masterRelayIndex: "\002"
  masterRelayOn: true
  indicesOfControlledRelaysOn: "\003"
  indicesOfControlledRelaysOff: "\003"
}
relayRefreshing: true
summerTimeDetails: "0360100"
winterTimeDetails: "1060200"
}

OSLP SetConfigurationResponse sent to platform:

setConfigurationResponse {
  status: OK
}
SetEventNotifications

SetEventNotifications messages

Description
Request which contains the EventNotification mask.
Response communicates status.

Message definitions

message SetEventNotificationsRequest {
  required uint32 NotificationMask = 1; // Bitmask for max 32 events, using NotificationBit
}

message SetEventNotificationsResponse {
  required Status status = 1;
}

Data types
enum Status {
  OK = 0;
  FAILURE = 1; // general failure
  REJECTED = 2; // request received in wrong state
}

Example
Soap requests and responses sent to and from platform:

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:com="http:
  <soapenv:Header>
    <com:OrganisationIdentification>LianderNetManagement</com:OrganisationIdentification>
  </soapenv:Header>
  <soapenv:Body>
    <dev:SetEventNotificationsRequest>
      <!--type: Identification-->
      <dev:DeviceIdentification>device-01</dev:DeviceIdentification>
      <dev:EventNotifications>DIAG_EVENTS</dev:EventNotifications>
      <dev:EventNotifications>HARDWARE_FAILURE</dev:EventNotifications>
      <dev:EventNotifications>LIGHT_EVENTS</dev:EventNotifications>
      <dev:EventNotifications>TARIFF_EVENTS</dev:EventNotifications>
      <dev:EventNotifications>MONITOR_EVENTS</dev:EventNotifications>
      <dev:EventNotifications>FIRMWARE_EVENTS</dev:EventNotifications>
      <dev:EventNotifications>SECURITY_EVENTS</dev:EventNotifications>
    </dev:SetEventNotificationsRequest>
  </soapenv:Body>
</soapenv:Envelope>

  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:SetEventNotificationsAsyncResponse xmlns:ns2="http://www.opensmartgridplatform.org">
      <ns2:AsyncResponse>
        <ns3:CorrelationUid>LianderNetManagement||device-01||20160104145052565</ns3:CorrelationUid>
        <ns3:DeviceId>device-01</ns3:DeviceId>
      </ns2:AsyncResponse>
    </ns2:SetEventNotificationsAsyncResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:com="http:
  <soapenv:Header>
    <com:OrganisationIdentification>LianderNetManagement</com:OrganisationIdentification>
  </soapenv:Header>
  <soapenv:Body>
    <!--type: Identification-->
    <com:UserName>liander gebruiker</com:UserName>
    <com:ApplicationName>WEB_NET_MANAGEMENT</com:ApplicationName>
  </soapenv:Body>
</soapenv:Envelope>

Grid eXchange Fabric Documentation
OSLP v0.6.1
OSLP SetEventNotificationsRequest sent to 'device-01' to set EventNotifications:

setEventNotificationsRequest {
  NotificationMask: 255
}

OSLP SetEventNotificationsResponse sent to platform:

setEventNotificationsResponse {
  status: OK
}
EventNotification

EventNotification messages

Description
Request sent from device to platform containing information about 1 to 6 events.
Response sent from platform to 'device-01' communicates status.

Message definitions

message EventNotificationRequest {
  repeated EventNotification notifications = 1; // [(nanopb).max_count = 6];
}

message EventNotificationResponse {
  required Status status = 1;
}

Datatypes

message EventNotification {
  required Event event = 1;
  optional bytes index = 2; // [(nanopb).max_size=1];
  optional string description = 3; // [(nanopb).max_size = 81];
  optional string timestamp = 4; // [(nanopb).max_size = 15]; // - Format YYYYMMDDhhmmss U
}

enum Status {
  OK = 0;
  FAILURE = 1; // general failure
  REJECTED = 2; // request received in wrong state
}

Example

OSLP request sent from 'device-01' to platform:

eventNotificationRequest {
  notifications {
    event: TARIFF_EVENTS_TARIFF_OFF
    index: "\001"
    description: "Tariff Off Example Event"
    timestamp: "20170404093500"
  }
}

OSLP response sent to 'device-01':

eventNotificationResponse {
  status: OK
}
SetSchedule

SetSchedule messages

Description
Request to set a light or tariff schedule on a device.
Response communicates status.

Message definitions

message SetScheduleRequest {
    repeated Schedule schedules = 1; // [(nanopb).max_count = 50];
    optional PageInfo pageInfo = 2;
    required RelayType scheduleType = 3; // RT_NOT_SET is NOT supported!
}

message SetScheduleResponse {
    required Status status = 1;
}

Datatypes

message Schedule {
    required Weekday weekday = 1;
    optional string startDay = 2; // [(nanopb).max_size = 9]; // Format YYYYMMDD UTC, indic
    optional string endDay = 3; // [(nanopb).max_size = 9]; // - Format YYYYMMDD UTC, includ
    required ActionTime actionTime = 4;
    optional string time = 5; // [(nanopb).max_size = 7]; // Format hmmss localtime set wth
    optional Window window = 6; // Window to wait for light sensor trigger.
    repeated LightValue value = 7; // [(nanopb).max_count = 6];
    optional TriggerType triggerType = 8; // React to setTransition or switch astronomical.
    optional uint32 minimumLightsOn = 9; // Minimal time (in seconds) the lights should bur,
    optional uint32 index = 10; // Index of schedule entry in the schedule list.
    optional bool isEnabled = 11; // Is this schedule entry enabled?
}

enum Weekday {
    MONDAY = 1;
    TUESDAY = 2;
    WEDNESDAY = 3;
    THURSDAY = 4;
    FRIDAY = 5;
    SATURDAY = 6;
    SUNDAY = 7;
    WEEKDAY = 8;
    WEEKEND = 9;
    ABSOLUTEDAY = 10;
    ALL = 11;
}

enum ActionTime {
    ABSOLUTETIME = 1;
    SUNRISE = 2;
    SUNSET = 3;
}

message Window {
    required uint32 minutesBefore = 1; // minutes before sunset / sunrise
    required uint32 minutesAfter = 2; // minutes after sunset / sunrise
}

message LightValue {
    optional bytes index = 1; // [(nanopb).max_size = 1]; // index number of connected light
    required bool on = 2;
    optional bytes dimValue = 3; // [(nanopb).max_size = 1]; // 1 - 100 %
}

enum TriggerType {
    TT_NOT_SET = 0;
}
LIGHT_TRIGGER = 1;
ASTRONOMICAL = 2;
}

message PageInfo {
  required uint32 currentPage = 1; // Pages start from 1
  required uint32 pageSize = 2;
  required uint32 totalPages = 3;
}

enum RelayType {
  RT_NOT_SET = 0;
  LIGHT = 1;
  TARIFF = 2;
}

enum Status {
  OK = 0;
  FAILURE = 1; // general failure
  REJECTED = 2; // request received in wrong state
}

Examples

Example 1: Light schedule based on light measurement

Screenshot of this schedule in an OSGP client application.

SOAP Request Message for Platform web service:

```xml
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:com="http://www.opensmartgridplatform.org/schemas/publiclighting/2014/10"
xmlns:sch="http://www.opensmartgridplatform.org/schemas/publiclighting/schedulemanagement/2014/10">
  <soapenv:Header>
    <com:OrganisationIdentification>LianderNetManagement</com:OrganisationIdentification>
  </soapenv:Header>
</soapenv:Envelope>
```
<soapenv:Header>
</soapenv:Header>

<soapenv:Body>

<s:SetScheduleRequest>
<!--type: Identification-->
<s:DeviceIdentification>device-01</s:DeviceIdentification>
<!--1 to 50 repetitions:--> 
<s:Schedules>
<!--type: WeekDayType - enumeration: [MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,-->
<s:WeekDay>All</s:WeekDay>
<!--type: ActionTimeType - enumeration: [ABSOLUTETIME,SUNRISE,SUNSET]--> 
<s:ActionTime>SUNRISE</s:ActionTime>
<!--Optional:--> 
<s:TriggerWindow>
<!--type: long--> 
<s:minutesBefore>15</s:minutesBefore>
<!--type: long--> 
<s:minutesAfter>15</s:minutesAfter>
</s:TriggerWindow>
<!--1 to 6 repetitions:--> 
<s:LightValue>
<!--Optional:--> 
<!--anonymous type--> 
<s:Index>0</s:Index>
<!--type: boolean--> 
<s:On>false</s:On>
</s:LightValue>
<!--Optional:--> 
<!--type: TriggerType - enumeration: [LIGHT_TRIGGER,ASTRONOMICAL]--> 
<s:TriggerType>LIGHT_TRIGGER</s:TriggerType>
</s:Schedules>

<s:Schedules>
<!--type: WeekDayType - enumeration: [MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,-->
<s:WeekDay>All</s:WeekDay>
<!--type: ActionTimeType - enumeration: [ABSOLUTETIME,SUNRISE,SUNSET]--> 
<s:ActionTime>SUNSET</s:ActionTime>
<!--Optional:--> 
<s:TriggerWindow>
<!--type: long--> 
<s:minutesBefore>15</s:minutesBefore>
<!--type: long--> 
<s:minutesAfter>15</s:minutesAfter>
</s:TriggerWindow>
<!--1 to 6 repetitions:--> 
<s:LightValue>
<!--Optional:--> 
<!--anonymous type--> 
<s:Index>0</s:Index>
<!--type: boolean--> 
<s:On>true</s:On>
</s:LightValue>
<!--Optional:--> 
<!--type: TriggerType - enumeration: [LIGHT_TRIGGER,ASTRONOMICAL]--> 
<s:TriggerType>LIGHT_TRIGGER</s:TriggerType>
</s:Schedules>

<s:Schedules>
<!--type: WeekDayType - enumeration: [MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,-->
<s:WeekDay>All</s:WeekDay>
<!--type: ActionTimeType - enumeration: [ABSOLUTETIME,SUNRISE,SUNSET]--> 
<s:ActionTime>ABSOLUTETIME</s:ActionTime>
<!--Optional:--> 
<!--type: string--> 
<s:Time>23:00:00</s:Time>
<!--Optional:--> 
<s:TriggerWindow>
<!--type: long--> 
<s:minutesBefore>30</s:minutesBefore>
<!--type: long--> 
</s:TriggerWindow>
</s:Schedules>

</s:SetScheduleRequest>
</soapenv:Body>
<sch:minutesAfter>30</sch:minutesAfter>
</sch:TriggerWindow>
<!-1 to 6 repetitions:->
<sch:LightValue>
<!-Optional:->
<!-anonymous type->
<sch:Index>2</sch:Index>
<!-type: boolean->
<sch:On>false</sch:On>
</sch:LightValue>
</sch:Schedules>

<sch:Schedules>
<!-type: WeekDayType - enumeration: [MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,:>
<sch:WeekDay>ALL</sch:WeekDay>
<!-type: ActionTimeType - enumeration: [ABSOLUTETIME,SUNRISE,SUNSET]-->>
<sch:ActionTime>ABSOLUTETIME</sch:ActionTime>
<!-Optional:->
<!-type: string->
<sch:Time>07:00:00</sch:Time>
<!-Optional:->
<sch:TriggerWindow>
<!-type: long->
<sch:minutesBefore>150</sch:minutesBefore>
<!-type: long->
<sch:minutesAfter>41</sch:minutesAfter>
</sch:TriggerWindow>
<!-1 to 6 repetitions:->
<sch:LightValue>
<!-Optional:->
<!-anonymous type->
<sch:Index>2</sch:Index>
<!-type: boolean->
<sch:On>true</sch:On>
<!-Optional:->
<!-anonymous type->
<!-<sch:DimValue>100</sch:DimValue>->
</sch:LightValue>
</sch:Schedules>
</sch:SetScheduleRequest>
</soapenv:Body>
</soapenv:Envelope>

SOAP Response Message:

  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns3:SetScheduleAsyncResponse xmlns:ns2="http://www.opensmartgridplatform.org/schemas/
      <ns3:SetScheduleAsyncResponse xmlns:ns2="http://www.opensmartgridplatform.org/schemas/
        <ns2:CorrelationUid>LianderNetManagement||device-01||20151230104608559</ns2:CorrelationUid>
        <ns2:DeviceId>device-01</ns2:DeviceId>
      </ns3:CorrelationUid>
    </ns3:SetScheduleAsyncResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

OSLP SetScheduleRequest sent to `device-01` to set a Light Schedule (1 page in this case, therefore no pagingInfo needed):

```xml
setScheduleRequest {
  schedules {
    weekday: ALL
    actionTime: SUNRISE
    window {
      minutesBefore: 15
      minutesAfter: 15
    }
    value {
      index: "\000"
      on: false
    }
```
triggerType: LIGHT_TRIGGER
}
schedules {
    weekday: ALL
    actionTime: SUNSET
    window {
        minutesBefore: 15
        minutesAfter: 15
    }
    value {
        index: "\000"
        on: true
    }
    triggerType: LIGHT_TRIGGER
}
schedules {
    weekday: ALL
    actionTime: ABSOLUTETIME
time: "230000"
    window {
        minutesBefore: 30
        minutesAfter: 30
    }
    value {
        index: "\002"
        on: false
    }
}
schedules {
    weekday: ALL
    actionTime: ABSOLUTETIME
time: "070000"
    window {
        minutesBefore: 150
        minutesAfter: 41
    }
    value {
        index: "\002"
        on: true
    }
}
scheduleType: LIGHT
}

OSLP SetScheduleResponse from 'device-01':

setScheduleResponse {
    status: OK
}

Description for this schedule:

This schedule combines a 'morning/evening light' with an 'all night light'. Relay 1 and 2 will be switched on using a light measurement trigger. Relay 2 will be switched off at 23:00 using an absolute time. Relay 2 will be switched on at 07:00, but only when no light measurement trigger has been received yet. Relay 1 and 2 will be switched off using a light measurement trigger.

The first schedule-entry:

schedules {
    weekday: ALL
    actionTime: SUNRISE
    window {
        minutesBefore: 15
        minutesAfter: 15
    }
    value {
        index: "\000"
        on: false
    }
    triggerType: LIGHT_TRIGGER
}
Definitions:

- "index: "000"" means: all device relays configured as LIGHT relays (see SetConfigurationRequest message)
- "light measurement trigger" is defined as: a SetTransitionRequest message containing a TransitionType matching the
  schedule-entry's actionTime value (SUNRISE matches NIGHT_DAY and SUNSET matches DAY_NIGHT)

Specifies: For all (weekday: ALL) 7 days of the week, when a light measurement trigger is received in the morning
  (actionTime: SUNRISE), then all device relays configured as LIGHT relays have to switch off (on: false).

When and only when a SUNRISE transition is received via a light measurement trigger (LIGHT_TRIGGER) within a window of
15 minutesBefore and 15 minutesAfter the calculated astronomical time for sunrise, then the device shall switch for the
received light measurement trigger.

When no SUNRISE transition is received via a light measurement trigger (LIGHT_TRIGGER) within a window of 15
minutesBefore and 15 minutesAfter the calculated astronomical time for sunrise, then the device shall switch at the end of the
window.

The triggerType (LIGHT_TRIGGER) defines how a SUNRISE (actionTime) transition will be triggered.

The second schedule-entry:

```
schedules {
  weekday: ALL
  actionTime: SUNSET
  window {
    minutesBefore: 15
    minutesAfter: 15
  }
  value {
    index: "\000"
    on: true
  }
  triggerType: LIGHT_TRIGGER
}
```

Definitions:

- "index: "000"" means: all device relays configured as LIGHT relays (see SetConfigurationRequest message)
- "light measurement trigger" is defined as: a SetTransitionRequest message containing a TransitionType matching the
  schedule-entry's actionTime value (SUNRISE matches NIGHT_DAY and SUNSET matches DAY_NIGHT)

Specifies: For all (weekday: ALL) 7 days of the week, when a light measurement trigger is received in the morning
  (actionTime: SUNSET), then all device relays configured as LIGHT relays have to switch on (on: true).

When and only when a SUNSET transition is received via a light measurement trigger (triggerType: LIGHT_TRIGGER) within
a window of 15 minutesBefore and 15 minutesAfter the calculated astronomical time for sunset, then the device shall switch
for the received light measurement trigger.

When no SUNSET transition is received via a light measurement trigger (triggerType: LIGHT_TRIGGER) within a window of
15 minutesBefore and 15 minutesAfter the calculated astronomical time for sunset, then the device shall switch at the end of the
window.

The triggerType (LIGHT_TRIGGER) defines how a SUNSET (actionTime) transition will be triggered.

The third schedule-entry:

```
schedules {
  weekday: ALL
  actionTime: ABSOLUTETIME
  time: "230000"
  window {
    minutesBefore: 30
    minutesAfter: 30
  }
  value {
    index: "\002"
    on: false
  }
}
```

Specifies: For all (weekday: ALL) 7 days of the week, when its 11 'o clock in the evening (actionTime: ABSOLUTETIME and
time: "230000") then device relay 2 has to switch off (on: false).

Since actionTime is ABSOLUTETIME, the triggerType value must be omitted from this schedule-entry.
The fourth schedule-entry:

```plaintext
schedules {
  weekday: ALL
  actionTime: ABSOLUTETIME
  time: "070000"
  window {
    minutesBefore: 150
    minutesAfter: 41
  }
  value {
    index: "\002"
    on: true
  }
}
```

For all (weekday: ALL) 7 days of the week, when its 7 'o clock in the morning (actionTime: ABSOLUTETIME and time: "070000") and there are no other schedule-entries that have caused the switching of device relay 2 within the window defined (minutesBefore: 150 and minutesAfter) then device relay 2 has to switch on (on: true).

Since actionTime is ABSOLUTETIME, the triggerType value must be omitted from this schedule-entry.

The last element of the SetScheduleRequest:

```
scheduleType: LIGHT
```

specifies that this is a light schedule.

SOAP Request to obtain response from 'device-01':

```xml
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
  xmlns:com=http://www.opensmartgridplatform.org/schemas/common/2014/10"
  xmlns:sch=http://www.opensmartgridplatform.org/schemas/publiclighting/schedulemanagement/2014/10">
  <soapenv:Header>
    <com:OrganisationIdentification>LianderNetManagement</com:OrganisationIdentification>
    <com:UserName>Kevin</com:UserName>
    <com:ApplicationName>SoapUI</com:ApplicationName>
  </soapenv:Header>
  <soapenv:Body>
    <sch:SetScheduleAsyncRequest>
      <sch:AsyncRequest>
        <com:CorrelationUid>LianderNetManagement|||device-01|||20151230104608559</com:CorrelationUid>
        <com:DeviceId>device-01</com:DeviceId>
      </sch:AsyncRequest>
    </sch:SetScheduleAsyncRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

SOAP Response containing response from 'device-01':

```xml
  xmlns:ns3=http://www.opensmartgridplatform.org/schemas/publiclighting/schedulemanagement/2014/10"
  xmlns:ns2=http://www.opensmartgridplatform.org/schemas/common/2014/10">
  <ns3:SetScheduleResponse>
    <ns3:Result>OK</ns3:Result>
  </ns3:SetScheduleResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example 2: Light schedule based on absolute time and day

SOAP messages:

```xml
  xmlns:ns3=http://www.opensmartgridplatform.org/schemas/publiclighting/schedulemanagement/2014/10"
  xmlns:ns2=http://www.opensmartgridplatform.org/schemas/common/2014/10">
  <ns3:SetScheduleRequest>
    <ns3:Result>OK</ns3:Result>
  </ns3:SetScheduleResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
OSLP SetScheduleRequest sent to 'device-01' to set a Light Schedule:

setScheduleRequest {
  schedules {
    weekday: ABSOLUTEDAY
    startDay: "2016-01-01"
    actionTime: ABSOLUTETIME
    time: "07:00:00"
    value {
      index: "001"
      on: false
    }
  }
  scheduleType: LIGHT
}

OSLP SetScheduleResponse sent to platform:

setScheduleResponse {
  status: OK
}

OSLP v0.6.1
This schedule has one entry which switches light relay 1 (index: "001") off at January 1st 2016 at 7 'o clock in the morning. When 'weekday' is set to ABSOLUTEDAY, the date will be placed in 'startDay'.

Example 3: Schedule using OSLP v0.6.1 specific properties

SOAP messages:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:com="http://www.opensmartgridplatform.org/schemas/publiclighting/2014/10"
                  xmlns:sch="http://www.opensmartgridplatform.org/schemas/publiclighting/schedulemanagement/2014/10">
    <com:OrganisationIdentification>LianderNetManagement</com:OrganisationIdentification>
    <com:UserName>Kevin</com:UserName>
    <com:ApplicationName>SoapUI</com:ApplicationName>
    <soapenv:Body>
        <sch:SetScheduleRequest>
            <sch:DeviceIdentification>device-01</sch:DeviceIdentification>
            <sch:Schedules>
                <sch:WeekDay>ALL</sch:WeekDay>
                <sch:ActionTime>SUNRISE</sch:ActionTime>
                <sch:TriggerWindow>
                    <sch:minutesBefore>15</sch:minutesBefore>
                    <sch:minutesAfter>15</sch:minutesAfter>
                </sch:TriggerWindow>
                <sch:LightValue>
                    <sch:Index>0</sch:Index>
                    <sch:On>false</sch:On>
                </sch:LightValue>
                <sch:TriggerType>LIGHT_TRIGGER</sch:TriggerType>
                <sch:Index>0</sch:Index>
                <sch:IsEnabled>true</sch:IsEnabled>
                <sch:minimumLightsOn>300</sch:minimumLightsOn>
            </sch:Schedules>
        </sch:SetScheduleRequest>
    </soapenv:Body>
</soapenv:Envelope>
```
<!-anonymous type-->  
<!-type: boolean-->  
<!-type: int, index of this schedule-entry-->  
<!-type: boolean-->  
<!-type: int, minimal burning time in seconds-->  
<!-type: WeekDayType - enumeration: [MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,:>
<!-type: ActionTimeType - enumeration: [ABSOLUTETIME,SUNRISE,SUNSET]->
<!-type: string-->  
<!-type: long-->  
<!-type: long-->  
<!-type: WeekDayType - enumeration: [MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,:>
<!-type: ActionTimeType - enumeration: [ABSOLUTETIME,SUNRISE,SUNSET]->
<!-type: string-->  
<!-type: long-->  
<!-type: long-->  
<!-type: anonymous type-->  
<!-type: boolean-->  
<!-type: int, index of this schedule-entry-->  
<!-type: boolean-->  
<!-type: int, minimal burning time in seconds-->  
<!-type: WeekDayType - enumeration: [MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,:>
<!-type: ActionTimeType - enumeration: [ABSOLUTETIME,SUNRISE,SUNSET]->
<!-type: string-->  
<!-type: long-->  
<!-type: long-->  
<!-type: anonymous type-->  
<!-type: boolean-->  
<!-type: int, index of this schedule-entry-->  
<!-type: boolean-->  
<!-type: int, minimal burning time in seconds-->  
<!-type: WeekDayType - enumeration: [MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,:>
<!-type: ActionTimeType - enumeration: [ABSOLUTETIME,SUNRISE,SUNSET]->
<!-type: string-->  
<!-type: long-->  
<!-type: long-->  
<!-type: anonymous type-->  
<!-type: boolean-->  
<!-type: int, index of this schedule-entry-->  
<!-type: boolean-->  
<!-type: int, minimal burning time in seconds-->  
<!-type: WeekDayType - enumeration: [MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,:>
<!-type: ActionTimeType - enumeration: [ABSOLUTETIME,SUNRISE,SUNSET]->
<!-type: string-->  
<!-type: long-->  
<!-type: long-->  
<!-type: anonymous type-->  
<!-type: boolean-->  
<!-type: int, index of this schedule-entry-->  
<!-type: boolean-->  
<!-type: int, minimal burning time in seconds-->  
<!-type: WeekDayType - enumeration: [MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,:>
<!-type: ActionTimeType - enumeration: [ABSOLUTETIME,SUNRISE,SUNSET]->
<!-type: string-->  
<!-type: long-->  
<!-type: long-->  
<!-type: anonymous type-->  
<!-type: boolean-->  
<!-type: int, index of this schedule-entry-->  
<!-type: boolean-->  
<!-type: int, minimal burning time in seconds-->  
<!-type: WeekDayType - enumeration: [MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,:>
<!-type: ActionTimeType - enumeration: [ABSOLUTETIME,SUNRISE,SUNSET]->
<!-type: string-->  
<!-type: long-->  
<!-type: long-->  
<!-type: anonymous type-->  
<!-type: boolean-->  
<!-type: int, index of this schedule-entry-->  
<!-type: boolean-->  
<!-type: int, minimal burning time in seconds-->  
<!-type: WeekDayType - enumeration: [MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,:>
<!-type: ActionTimeType - enumeration: [ABSOLUTETIME,SUNRISE,SUNSET]->
<!-type: string-->  
<!-type: long-->  
<!-type: long-->  
<!-type: anonymous type-->  
<!-type: boolean-->  
<!-type: int, index of this schedule-entry-->  
<!-type: boolean-->  
<!-type: int, minimal burning time in seconds-->  
<!-type: WeekDayType - enumeration: [MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,:>
<!-type: ActionTimeType - enumeration: [ABSOLUTETIME,SUNRISE,SUNSET]->
<!-type: string-->  
<!-type: long-->  
<!-type: long-->  
<!-type: anonymous type-->  
<!-type: boolean-->  
<!-type: int, index of this schedule-entry-->  
<!-type: boolean-->  
<!-type: int, minimal burning time in seconds-->  
<!-type: WeekDayType - enumeration: [MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,:>
<!-type: ActionTimeType - enumeration: [ABSOLUTETIME,SUNRISE,SUNSET]->
<!-type: string-->  
<!-type: long-->  
<!-type: long-->  
<!-type: anonymous type-->  
<!-type: boolean-->  
<!-type: int, index of this schedule-entry-->  
<!-type: boolean-->  
<!-type: int, minimal burning time in seconds-->  
<!-type: WeekDayType - enumeration: [MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,:>
<!-type: ActionTimeType - enumeration: [ABSOLUTETIME,SUNRISE,SUNSET]->
<!-type: string-->  
<!-type: long-->  
<!-type: long-->  
<!-type: anonymous type-->  
<!-type: boolean-->  
<!-type: int, index of this schedule-entry-->  
<!-type: boolean-->  
<!-type: int, minimal burning time in seconds-->  
<!-type: WeekDayType - enumeration: [MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,:>
<sch:Index>1</sch:Index>
<!- type: boolean -->
<sch:On>true</sch:On>
<!- Optional: -->
<!- anonymous type -->
<!- type: int, index of this schedule-entry -->
<sch:Index>3</sch:Index>
<!- Optional: -->
<!- type: boolean -->
<sch:IsEnabled>true</sch:IsEnabled>
<!- Optional: -->
<!- type: int, minimal burning time in seconds -->
<sch:minimumLightsOn>300</sch:minimumLightsOn>
</sch:Schedules>
</sch:SetScheduleRequest>
</soapenv:Body>
</soapenv:Envelope>

  <SOAP-ENV:Body>
      <ns3:CorrelationUid>LianderNetManagement|||device-01|||20160313162236547</ns3:CorrelationUid>
      <ns3:DeviceId>device-01</ns3:DeviceId>
    </ns2:SetScheduleAsyncResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

OSLP SetScheduleRequest sent to 'device-01':

setScheduleRequest {
  schedules {
    weekday: ALL
    actionTime: SUNRISE
    window {
      minutesBefore: 15
      minutesAfter: 15
    }
    value {
      index: "\000"
      on: false
    }
  }
}
triggerType: LIGHT_TRIGGER
index: 0
isEnabled: true
}
schedules {
  weekday: ALL
  actionTime: SUNSET
  window {
    minutesBefore: 15
    minutesAfter: 15
  }
  value {
    index: "\000"
    on: true
  }
  triggerType: LIGHT_TRIGGER
  index: 1
  isEnabled: true
}
schedules {
  weekday: ALL
  actionTime: ABSOLUTETIME
time: "230000"
  window {
    minutesBefore: 30
    minutesAfter: 30
  }
  value {
    index: "\001"
    on: false
  }
  index: 2
  isEnabled: true
}
schedules {
  weekday: ALL
  actionTime: ABSOLUTETIME
time: "070000"
  window {
    minutesBefore: 30
    minutesAfter: 30
  }
  value {
    index: "\001"
    on: true
  }
  minimumLightsOn: 300
  index: 3
  isEnabled: true
}
scheduleType: LIGHT
}

OSLP SetScheduleResponse from 'device-01':

setScheduleResponse {
  status: OK
}

Description for this schedule:

This schedule consists of 1 page, and uses 'minimumLightsOn' to indicate a minimal burning time in seconds. Further it uses 'index' and 'isEnabled' variables for the Schedule struct, to indicate what index this schedule-entry has within the list of schedule-entries and whether or not the schedule-entry is enabled.

Astronomical Offsets

The SOAP request message may contain information about astronomical offsets (see the documentation about light schedules for more details about the offsets). When AstronomicalSunriseOffset and/or AstronomicalSunsetOffset are set, they will be configured on the device by updating the configuration setting the offsets as astroGateSunRiseOffset and astroGateSunSetOffset of the SetConfigurationRequest.
Example 4: Tariff Schedule

SOAP Request Message for Platform web service:

```xml
    xmlns:sch="http://schemas.opengridhub.org/2014/10">
  <soapenv:Header>
    <com:OrganisationIdentification>LianderNetManagement</com:OrganisationIdentification>
    <com:UserName>Kevin</com:UserName>
    <com:ApplicationName>SoapUI</com:ApplicationName>
  </soapenv:Header>
  <soapenv:Body>
    <sch:SetScheduleRequest>
      <sch:DeviceIdentification>device-01</sch:DeviceIdentification>
      <!--1 to 50 repetitions:-->
      <sch:Schedules>
        <sch:WeekDay>WEEKDAY</sch:WeekDay>
        <sch:StartDay>2015-01-01</sch:StartDay>
        <sch:EndDay>2016-02-01</sch:EndDay>
        <sch:Time>23:00:00</sch:Time>
        <!--1 to 6 repetitions:-->
        <sch:TariffValue>
          <sch:Index>3</sch:Index>
          <sch:High>0</sch:High>
        </sch:TariffValue>
      </sch:Schedules>
      <sch:Schedules>
        <sch:WeekDay>WEEKDAY</sch:WeekDay>
        <sch:StartDay>2015-01-01</sch:StartDay>
        <sch:EndDay>2016-02-01</sch:EndDay>
        <sch:Time>07:00:00</sch:Time>
        <!--1 to 6 repetitions:-->
        <sch:TariffValue>
          <sch:Index>3</sch:Index>
          <sch:High>1</sch:High>
        </sch:TariffValue>
      </sch:Schedules>
      <sch:Schedules>
        <sch:WeekDay>ABSOLUTEDAY</sch:WeekDay>
        <sch:StartDay>2015-01-01</sch:StartDay>
        <sch:EndDay>2015-01-01</sch:EndDay>
        <sch:Time>07:00:00</sch:Time>
        <!--1 to 6 repetitions:-->
        <sch:TariffValue>
          <sch:Index>3</sch:Index>
          <sch:High>0</sch:High>
        </sch:TariffValue>
      </sch:Schedules>
      <sch:Schedules>
        <sch:WeekDay>ABSOLUTEDAY</sch:WeekDay>
        <sch:StartDay>2015-04-06</sch:StartDay>
        <sch:EndDay>2015-04-06</sch:EndDay>
        <sch:Time>07:00:00</sch:Time>
        <!--1 to 6 repetitions:-->
        <sch:TariffValue>
          <sch:Index>3</sch:Index>
          <sch:High>0</sch:High>
        </sch:TariffValue>
      </sch:Schedules>
      <sch:Schedules>
        <sch:WeekDay>ABSOLUTEDAY</sch:WeekDay>
        <sch:EndDay>2015-04-27</sch:EndDay>
        <sch:Time>07:00:00</sch:Time>
        <!--1 to 6 repetitions:-->
        <sch:TariffValue>
          <sch:Index>3</sch:Index>
          <sch:High>0</sch:High>
        </sch:TariffValue>
      </sch:Schedules>
    </sch:SetScheduleRequest>
  </soapenv:Body>
</soapenv:Envelope>
```
<soapenv:Body>


<soapenv:Body>

<Grid eXchange Fabric Documentation>

<sch:SetScheduleRequest>

<sch:Schedules>

<sch:Schedules>

<sch:Schedules>

<sch:Schedules>

<sch:Schedules>

<sch:Schedules>

<sch:Schedules>

<sch:Schedules>

<sch:Schedules>

<sch:Schedules>

<sch:Schedules>

<sch:Schedules>

<sch:Schedules>

<sch:Schedules>

<sch:Schedules>

<sch:Schedules>

<sch:Schedules>

<sch:Schedules>

</sch:Schedules>

</sch:Schedules>

</sch:Schedules>

</sch:Schedules>

</sch:Schedules>

</sch:Schedules>

</soapenv:Body>
</soapenv:Envelope>

SOAP Response Message:


<SOAP-ENV:Header/>
</SOAP-ENV:Body>

OSLP v0.6.1
OSLP SetScheduleRequest sent to 'device-01' to set a Tariff Schedule (2 pages in this case):

```xml
setScheduleRequest {
    schedules {
        weekday: WEEKDAY
        startDay: "20150101"
        endDay: "20160201"
        actionTime: ABSOLUTETIME
        time: "230000"
        value {
            index: "\003"
            on: true
        }
    }
    schedules {
        weekday: WEEKDAY
        startDay: "20150101"
        endDay: "20160201"
        actionTime: ABSOLUTETIME
        time: "070000"
        value {
            index: "\003"
            on: false
        }
    }
    schedules {
        weekday: ABSOLUTEDAY
        startDay: "20150101"
        endDay: "20150101"
        actionTime: ABSOLUTETIME
        time: "070000"
        value {
            index: "\003"
            on: true
        }
    }
    schedules {
        weekday: ABSOLUTEDAY
        startDay: "20150406"
        endDay: "20150406"
        actionTime: ABSOLUTETIME
        time: "070000"
        value {
            index: "\003"
            on: true
        }
    }
    schedules {
        weekday: ABSOLUTEDAY
        startDay: "20150427"
        endDay: "20150427"
        actionTime: ABSOLUTETIME
        time: "070000"
        value {
            index: "\003"
            on: true
        }
    }
}
pageInfo {
    currentPage: 1
    pageSize: 5
    totalPages: 2
}
```
OSLP SetScheduleResponse from 'device-01' for page 1:

setScheduleResponse {
  status: OK
}

setScheduleRequest {
  schedules {
    weekday: ABSOLUTEDAY
    startDay: "20150514"
    endDay: "20150514"
    actionTime: ABSOLUTETIME
    time: "070000"
    value {
      index: "\003"
      on: true
    }
  }
  schedules {
    weekday: ABSOLUTEDAY
    startDay: "20150525"
    endDay: "20150525"
    actionTime: ABSOLUTETIME
    time: "070000"
    value {
      index: "\003"
      on: true
    }
  }
  schedules {
    weekday: ABSOLUTEDAY
    startDay: "20151225"
    endDay: "20151225"
    actionTime: ABSOLUTETIME
    time: "070000"
    value {
      index: "\003"
      on: true
    }
  }
  schedules {
    weekday: ABSOLUTEDAY
    startDay: "20151226"
    endDay: "20151226"
    actionTime: ABSOLUTETIME
    time: "070000"
    value {
      index: "\003"
      on: true
    }
  }
  schedules {
    weekday: ABSOLUTEDAY
    startDay: "20160101"
    endDay: "20160101"
    actionTime: ABSOLUTETIME
    time: "070000"
    value {
      index: "\003"
      on: true
    }
  }
}

pageInfo {
  currentPage: 2
  pageSize: 5
  totalPages: 2
}

scheduleType: TARIFF

OSLP v0.6.1 489
OSLP SetScheduleResponse from 'device-01' for page 2:

```
setScheduleResponse {
  status: OK
}
```

Description for this schedule:

This schedule defines the tariff switching moments. For most weekdays of the year the tariff is high from 7 ‘o clock in the morning until 11 ‘o clock in the evening. During the night and weekend, the tariff is low. However for certain days, like Christmas Day, the tariff has to be low as well (Christmas Day may be a weekday).

The first schedule-entry:

```
schedules {
  weekday: WEEKDAY
  startDay: "20150101"
  endDay: "20160201"
  actionTime: ABSOLUTETIME
  time: "230000"
  value {
    index: "\003"
    on: true
  }
}
```

specifies that for every work day of the week (weekday: WEEKDAY meaning from Monday until Friday) from 1st of January 2015 until 1st of February 2016 (startDay: "20150101" and endDay: "20160201") at 11 ‘o clock in the evening (actionTime: ABSOLUTETIME and time: "230000") the relay with index 3 (index: "\003") has to switch on (on: true). When a device is configured to have relay 3 as TARIFF relay, this means the tariff will be low. When a device is configured to have relay 3 as TARIFF_REVERSED, this means the tariff will be high.

The second schedule-entry:

```
schedules {
  weekday: WEEKDAY
  startDay: "20150101"
  endDay: "20160201"
  actionTime: ABSOLUTETIME
  time: "070000"
  value {
    index: "\003"
    on: false
  }
}
```

specifies that for every work day of the week (weekday: WEEKDAY meaning from Monday until Friday) from 1st of January 2015 until 1st of February 2016 (startDay: "20150101" and endDay: "20160201") at 7 ‘o clock in the morning (actionTime: ABSOLUTETIME and time: "070000") the relay with index 3 (index: "\003") has to switch off (on: false). When a device is configured to have relay 3 as TARIFF relay, this means the tariff will be high. When a device is configured to have relay 3 as TARIFF_REVERSED, this means the tariff will be low.

The third schedule-entry:

```
schedules {
  weekday: WEEKDAY
  startDay: "20150101"
  endDay: "20150101"
  actionTime: ABSOLUTETIME
  time: "070000"
  value {
    index: "\003"
    on: true
  }
}
```

specifies that for every work day of the week (weekday: WEEKDAY meaning from Monday until Friday) from 1st of January 2015 until 1st of January 2015 (startDay: "20150101" and endDay: "20150101") at 7 ‘o clock in the morning (actionTime: ABSOLUTETIME and time: "070000") the relay with index 3 (index: "\003") has to switch on (on: true). When a device is configured to have relay 3 as TARIFF relay, this means the tariff will be low. When a device is configured to have relay 3 as TARIFF_REVERSED, this means the tariff will be high. This schedule entry is needed to make sure that the tariff is low for a particular day of the year (New Year's Day).

The fourth schedule-entry:
specifies that for every work day of the week (weekday: WEEKDAY meaning from Monday until Friday) from 6th of April 2015 until 6th of April 2015 (startDay: "20150406" and endDay: "20150406") at 7 'o clock in the morning (actionTime: ABSOLUTETIME and time: "070000") the relay with index 3 (index: "\003") has to switch on (on: true). When a device is configured to have relay 3 as TARIFF relay, this means the tariff will be low. When a device is configured to have relay 3 as TARIFF_REVERSED, this means the tariff will be high. This schedule entry is needed to make sure that the tariff is low for a particular day of the year (Easter Monday).

The fifth schedule-entry:

```plaintext
schedules {
    weekday: WEEKDAY
    startDay: "20150427"
    endDay: "20150427"
    actionTime: ABSOLUTETIME
    time: "070000"
    value {
        index: "\003"
        on: true
    }
}
```

specifies that for every work day of the week (weekday: WEEKDAY meaning from Monday until Friday) from 27th of April 2015 until 27th of April 2015 (startDay: "20150427" and endDay: "20150427") at 7 'o clock in the morning (actionTime: ABSOLUTETIME and time: "070000") the relay with index 3 (index: "\003") has to switch on (on: true). When a device is configured to have relay 3 as TARIFF relay, this means the tariff will be low. When a device is configured to have relay 3 as TARIFF_REVERSED, this means the tariff will be high. This schedule entry is needed to make sure that the tariff is low for a particular day of the year (Dutch Kings Day).

The pagination info:

```plaintext
pageInfo {
    currentPage: 1
    pageSize: 5
    totalPages: 2
}
```

specifies that this is the first page of a total of 2 pages. The pageSize is set by the platform and can be any value from 1 to 50.

The last element of the SetScheduleRequest:

```plaintext
scheduleType: TARIFF
```

specifies that this is a tariff schedule.

The sixth schedule-entry (page 2):

```plaintext
schedules {
    weekday: WEEKDAY
    startDay: "20150514"
    endDay: "20150514"
    actionTime: ABSOLUTETIME
    time: "070000"
    value {
        index: "\003"
        on: true
    }
}
```

specifies that for every work day of the week (weekday: WEEKDAY meaning from Monday until Friday) from 14th of May 2015 until 14th of May 2015 (startDay: "20150514" and endDay: "20150514") at 7 'o clock in the morning (actionTime: ABSOLUTETIME and time: "070000") the relay with index 3 (index: "\003") has to switch on (on: true). When a device is configured to have relay 3 as TARIFF relay, this means the tariff will be low. When a device is configured to have relay 3 as TARIFF_REVERSED, this means the tariff will be high. This schedule entry is needed to make sure that the tariff is low for a particular day of the year (Easter Monday).
TARIFF_REVERSED, this means the tariff will be high. This schedule entry is needed to make sure that the tariff is low for a particular day of the year (Ascension Day).

The seventh schedule-entry (page 2):

```plaintext
schedules {
    weekday: WEEKDAY
    startDay: "20150525"
    endDay: "20150525"
    actionTime: ABSOLUTETIME
    time: "070000"
    value {
        index: "\003"
        on: true
    }
}
```

specifies that for every work day of the week (weekday: WEEKDAY meaning from Monday until Friday) from 25th of May 2015 until 25th of May 2015 (startDay: "20150525" and endDay: "20150525") at 7 'o clock in the morning (actionTime: ABSOLUTETIME and time: "070000") the relay with index 3 (index: "003") has to switch on (on: true). When a device is configured to have relay 3 as TARIFF relay, this means the tariff will be low. When a device is configured to have relay 3 as TARIFF_REVERSED, this means the tariff will be high. This schedule entry is needed to make sure that the tariff is low for a particular day of the year (Whit Monday).

The eighth schedule-entry (page 2):

```plaintext
schedules {
    weekday: WEEKDAY
    startDay: "20151225"
    endDay: "20151225"
    actionTime: ABSOLUTETIME
    time: "070000"
    value {
        index: "\003"
        on: true
    }
}
```

specifies that for every work day of the week (weekday: WEEKDAY meaning from Monday until Friday) from 25th of December 2015 until 25th of December 2015 (startDay: "20151225" and endDay: "20151225") at 7 'o clock in the morning (actionTime: ABSOLUTETIME and time: "070000") the relay with index 3 (index: "003") has to switch on (on: true). When a device is configured to have relay 3 as TARIFF relay, this means the tariff will be low. When a device is configured to have relay 3 as TARIFF_REVERSED, this means the tariff will be high. This schedule entry is needed to make sure that the tariff is low for a particular day of the year (Christmas Day).

The ninth schedule-entry (page 2):

```plaintext
schedules {
    weekday: WEEKDAY
    startDay: "20160101"
    endDay: "20160101"
    actionTime: ABSOLUTETIME
    time: "070000"
    value {
        index: "\003"
        on: true
    }
}
```

specifies that for every work day of the week (weekday: WEEKDAY meaning from Monday until Friday) from 1st of January 2016 until 1st of January 2016 (startDay: "20160101" and endDay: "20160101") at 7 'o clock in the morning (actionTime: ABSOLUTETIME and time: "070000") the relay with index 3 (index: "003") has to switch on (on: true). When a device is configured to have relay 3 as TARIFF relay, this means the tariff will be low. When a device is configured to have relay 3 as TARIFF_REVERSED, this means the tariff will be high. This schedule entry is needed to make sure that the tariff is low for a particular day of the year (New Year's Day).

The pagination info (page 2):

```plaintext
pageInfo {
    currentPage: 2
    pageSize: 5
    totalPages: 2
}
```

specifies that this is the second page of a total of 2 pages. The pageSize is set by the platform and can be any value from 1 to
The last element of the SetScheduleRequest:

```
scheduleType: TARIFF
```

specifies that this is a tariff schedule.

SOAP Request to obtain response from 'device-01':

```

<soapenv:Header>
    <com:OrganisationIdentification>LianderNetManagement</com:OrganisationIdentification>
    <com:UserName>Kevin</com:UserName>
    <com:ApplicationName>SoapUI</com:ApplicationName>
</soapenv:Header>

<soapenv:Body>
    <sch:SetScheduleAsyncRequest>
        <sch:AsyncRequest>
            <com:CorrelationUid>LianderNetManagement|||device-01|||20151230132054477</com:CorrelationUid>
            <com:DeviceId>device-01</com:DeviceId>
        </sch:AsyncRequest>
    </sch:SetScheduleAsyncRequest>
</soapenv:Body>
</soapenv:Envelope>
```

SOAP Response containing response from 'device-01':

```

<SOAP-ENV:Header/>
<SOAP-ENV:Body>
    <ns3:SetScheduleResponse xmlns:ns2="http://www.opensmartgridplatform.org/schemas/common"
        <ns3:Result>OK</ns3:Result>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
ResumeSchedule

ResumeSchedule messages

Description
Request that notifies the device to continue the current schedule after the current schedule was interrupted (for example by switching by hand using SetLightRequest). This request can operate on a single relay or on all relays and the resuming of the schedule can be immediate or at the next schedule-entry.

Response which confirms the ResumeScheduleRequest has been executed or rejects the ResumeScheduleRequest.

Message definitions

message ResumeScheduleRequest {
  optional bytes index = 1; // [(nanopb).max_size = 1]; // index number of connected light
  required bool immediate = 2; // [default = false]; // Resume at next schedule item or dir
}

message ResumeScheduleResponse {
  required Status status = 1;
}

Data types

enum Status {
  OK = 0;
  FAILURE = 1; // general failure
  REJECTED = 2; // request received in wrong state
}

Example
Soap requests and responses sent to and from platform:

```xml
xmlns:adh="http://www.opensmartgridplatform.org/schemas/publiclighting/adhocmanagement/2014/10">
  <soapenv:Header>
    <com:OrganisationIdentification>LianderNetManagement</com:OrganisationIdentification>
    <com:UserName>liander gebruiker</com:UserName>
    <com:ApplicationName>WEB_NET_MANAGEMENT</com:ApplicationName>
  </soapenv:Header>
  <soapenv:Body>
    <adh:ResumeScheduleAsyncRequest>
      <adh:DeviceIdentification>device-01</adh:DeviceIdentification>
    </adh:ResumeScheduleAsyncRequest>
  </soapenv:Body>
</soapenv:Envelope>

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:com="http://www.opensmartgridplatform.org/schemas/publiclighting/2014/10"
xmlns:adh="http://www.opensmartgridplatform.org/schemas/publiclighting/adhocmanagement/2014/10">
  <soapenv:Header>
    <com:OrganisationIdentification>LianderNetManagement</com:OrganisationIdentification>
    <com:UserName>liander gebruiker</com:UserName>
    <com:ApplicationName>WEB_NET_MANAGEMENT</com:ApplicationName>
  </soapenv:Header>
  <soapenv:Body>
    <adh:ResumeScheduleRequest>
      <adh:DeviceIdentification>device-01</adh:DeviceIdentification>
      <adh:Index>1</adh:Index>
      <adh:IsImmediate>1</adh:IsImmediate>
    </adh:ResumeScheduleRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

OSLP v0.6.1 494
<adh:AsyncRequest>
  <com:CorrelationUid>LianderNetManagement|||device-01|||20160104152159539</com:CorrelationUid>
  <com:DeviceId>device-01</com:DeviceId>
</adh:AsyncRequest>
</soapenv:Body>
</soapenv:Envelope>

  xmlns:ns2="http://www.opensmartgridplatform.org/schemas/publiclighting/adhocmanagement/2014/10">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:ResumeScheduleResponse>
      <ns2:Result>OK</ns2:Result>
    </ns2:ResumeScheduleResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

**OSLP** ResumeScheduleRequest sent to 'device-01':

```javascript
resumeScheduleRequest {
  index: "\001"
  immediate: true
}
```

**OSLP** ResumeScheduleResponse sent to platform:

```javascript
resumeScheduleResponse {
  status: OK
}
```
GetFirmwareVersion

GetFirmwareVersion messages

Description
Request which notifies the device to send the current firmware version.
Response containing the firmware version.

Message definitions
message GetFirmwareVersionRequest {
  optional bool present = 1 [default = true];
}
message GetFirmwareVersionResponse {
  required string firmwareVersion = 1; // ((nanopb).max_size = 7); // RXX
}

Datatypes

Example
Soap requests and responses sent to and from platform:
  xmlns:fman="http://www.opensmartgridplatform.org/schemas/common/firmwaremanagement/2014/10"
  xmlns:ns2="http://www.opensmartgridplatform.org/schemas/common/2014/10"
  xmlns:ns3="http://www.opensmartgridplatform.org/schemas/common/2014/10">
  <soapenv:Header>
    <com:OrganisationIdentification>LianderNetManagement</com:OrganisationIdentification>
    <com:UserName>KevinSmeets</com:UserName>
    <com:ApplicationName>SoapUI</com:ApplicationName>
  </soapenv:Header>
  <soapenv:Body>
    <fman:GetFirmwareVersionRequest xmlns:fman="http://www.opensmartgridplatform.org/schemas/common/firmwaremanagement/2014/10">
      <fman:DeviceIdentification>device-01</fman:DeviceIdentification>
    </fman:GetFirmwareVersionRequest>
  </soapenv:Body>
</soapenv:Envelope>

  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:GetFirmwareVersionAsyncResponse xmlns:ns2="http://www.opensmartgridplatform.org/schemas/common/2014/10"
      xmlns:ns3="http://www.opensmartgridplatform.org/schemas/common/2014/10">  
      <ns3:CorrelationUid>LianderNetManagement||device-01||20160104150323405</ns3:CorrelationUid>
      <ns3:DeviceId>device-01</ns3:DeviceId>
    </ns2:GetFirmwareVersionAsyncResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

  xmlns:fman="http://www.opensmartgridplatform.org/schemas/common/firmwaremanagement/2014/10"
  xmlns:ns2="http://www.opensmartgridplatform.org/schemas/common/2014/10"
  xmlns:ns3="http://www.opensmartgridplatform.org/schemas/common/2014/10">  
  <soapenv:Header>
    <com:OrganisationIdentification>LianderNetManagement</com:OrganisationIdentification>
    <com:UserName>liander gebruiker</com:UserName>
    <com:ApplicationName>WEB_NET_MANAGEMENT</com:ApplicationName>
  </soapenv:Header>
  <soapenv:Body>
    <fman:AsyncRequest>
      <fman:DeviceIdentification>device-01</fman:DeviceIdentification>
    </fman:AsyncRequest>
  </soapenv:Body>
</soapenv:Envelope>
</soapenv:Envelope>

  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:GetFirmwareVersionResponse xmlns:ns2="http://www.opensmartgridplatform.org/schemas/common/firmwaremanagement/2014/10">
      <ns2:Result>OK</ns2:Result>
      <ns2:FirmwareVersion>R01</ns2:FirmwareVersion>
    </ns2:GetFirmwareVersionResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

OSLP GetFirmwareRequest message sent to 'device-01':
getFirmwareVersionRequest {
}

OSLP GetFirmwareResponse message sent to platform:
getFirmwareVersionResponse {
  firmwareVersion: "R01"
}
UpdateFirmware

UpdateFirmware messages

Description
Request for a device to download and install new firmware. The request contains a URL defining the location of the new firmware image. The device should download the firmware from that location.

Response communicates status.

Message definitions

message UpdateFirmwareRequest {
  required string firmwareDomain = 1;  // [(nanopb).max_size = 100]; // Servername
  required string firmwareUrl = 2;  // [(nanopb).max_size = 255]; // /firmware/TSTMAN/TSTMC
}

message UpdateFirmwareResponse {
  required Status status = 1;
}

Data types

enum Status {
  OK = 0;
  FAILURE = 1;  // general failure
  REJECTED = 2;  // request received in wrong state
}

Example
Soap requests and responses sent to and from platform:

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:com="http:
  <soapenv:Header>
    <com:OrganisationIdentification>LianderNetManagement</com:OrganisationIdentification>
    <com:UserName>liander gebruiker</com:UserName>
    <com:ApplicationName>WEB_NET_MANAGEMENT</com:ApplicationName>
  </soapenv:Header>
  <soapenv:Body>
    <fir1:UpdateFirmwareRequest xmlns:fir1="http://www.opensmartgridplatform.org/schemas/c
    <!--anonymous type-->
    <fir1:DeviceIdentification>device-01</fir1:DeviceIdentification>
    <fir1:FirmwareIdentification>TSTMAN/TSTMOD/SSLD-V17</fir1:FirmwareIdentification>
  </soapenv:Body>
</soapenv:Envelope>

  <ns2:AsyncResponse>
    <ns3:CorrelationUid>LianderNetManagement|||device-01|||20160104145959438</ns3:C
    <ns3:DeviceId>device-01</ns3:DeviceId>
  </ns2:AsyncResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:com="http:
  <soapenv:Header>
    <com:OrganisationIdentification>LianderNetManagement</com:OrganisationIdentification>
    <com:UserName>liander gebruiker</com:UserName>
    <com:ApplicationName>WEB_NET_MANAGEMENT</com:ApplicationName>
  </soapenv:Header>
  <soapenv:Body>
    <fir1:UpdateFirmwareAsyncRequest xmlns:fir1="http://www.opensmartgridplatform.org/sche
  </soapenv:Body>
</soapenv:Envelope>

OSLP v0.6.1
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<ns2:UpdateFirmwareResponse xmlns:ns2="http://www.opensmartgridplatform.org/schemas/common/firmwaremanagement/2014/10">
<ns2:Result>OK</ns2:Result>
</ns2:UpdateFirmwareResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

OSLP UpdateFirmwareRequest sent to 'device-01' to update firmware:

updateFirmwareRequest {
  firmwareDomain: "flexovltest.cloudapp.net"
  firmwareUrl: "/firmware/TSTMAN/TSTMOD/SSLD-V17.hex"
}

OSLP UpdateFirmwareResponse sent to the platform:

updateFirmwareResponse {
  status: OK
}
SetReboot

SetReboot messages

Description
Request which notifies the device to reboot immediately. After a reboot, the device will switch its relays according to its schedule. Any ad hoc changes to relays will be lost.

Response communicates status.

Message definitions
message SetRebootRequest {
  optional bool present = 1 [default = true];
}

message SetRebootResponse {
  required Status status = 1;
}

Data types
enum Status {
  OK = 0;
  FAILURE = 1; // general failure
  REJECTED = 2; // request received in wrong state
}

Example
Soap requests and responses sent to and from platform:

```xml
  xmlns:com="com">
  <soapenv:Header>
    <com:OrganisationIdentification>LianderNetManagement</com:OrganisationIdentification>
    <com:UserName>liander gebruiker</com:UserName>
    <com:ApplicationName>WEB_NET_MANAGEMENT</com:ApplicationName>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:SetRebootRequest>
      <ns1:DeviceIdentification>device-01</ns1:DeviceIdentification>
    </ns1:SetRebootRequest>
  </soapenv:Body>
</soapenv:Envelope>

  xmlns:ns2="http://www.opensmartgridplatform.org/schemas/common/2014/10"
  xmlns:ns3="http://www.opensmartgridplatform.org/schemas/common/2014/10">
  <SOAP-ENV:Header>
    <ns3:CorrelationUid>LianderNetManagement|||device-01|||20160104153201024</ns3:CorrelationUid>
    <ns2:DeviceId>device-01</ns2:DeviceId>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <ns3:SetRebootAsyncResponse>
      <ns3:AsyncResponse>
        <ns2:CorrelationUid>LianderNetManagement|||device-01|||20160104153201024</ns2:CorrelationUid>
        <ns2:DeviceId>device-01</ns2:DeviceId>
      </ns3:AsyncResponse>
    </ns3:SetRebootAsyncResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

  xmlns:com="com">
  <soapenv:Header>
    <com:OrganisationIdentification>LianderNetManagement</com:OrganisationIdentification>
    <com:UserName>liander gebruiker</com:UserName>
    <com:ApplicationName>WEB_NET_MANAGEMENT</com:ApplicationName>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:SetRebootAsyncRequest>
      <!--type: CorrelationUid-->
      <ns:CorrelationUid>LianderNetManagement|||device-01|||20160104153201024</ns:CorrelationUid>
      <!--type: Identification-->
      <ns1:DeviceIdentification>device-01</ns1:DeviceIdentification>
    </ns1:SetRebootAsyncRequest>
  </soapenv:Body>
</soapenv:Envelope>
```
<ns:DeviceId>device-01</ns:DeviceId>
</ns1:AsyncRequest>
</ns1:SetRebootAsyncRequest>
</soapenv:Body>
</soapenv:Envelope>

  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns3:SetRebootResponse xmlns:ns2="http://www.opensmartgridplatform.org/schemas/common">
      <ns3:Result>OK</ns3:Result>
    </ns3:SetRebootResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

**OSLP** SetRebootRequest message sent to 'device-01':

```xml
setRebootRequest {
}
```

**OSLP** SetRebootResponse sent to platform:

```xml
setRebootResponse {
  status: OK
}
```
StartSelfTest

StartSelfTest messages

Description
Request that notifies the device to switch all light relays on.

Response communicates status.

Message definitions

message StartSelfTestRequest {
  optional bool present = 1 [default = true];
}

message StartSelfTestResponse {
  required Status status = 1;
}

Data types

enum Status {
  OK = 0;
  FAILURE = 1; // general failure
  REJECTED = 2; // request received in wrong state
}

Example

Soap requests and responses sent to and from platform:

```xml
  xmlns:dev="http://www.opensmartgridplatform.org/schemas/deviceinstallation/2014/10"
>
  <soapenv:Header>
    <com:OrganisationIdentification>LianderNetManagement</com:OrganisationIdentification>
    <com:UserName>liander gebruiker</com:UserName>
    <com:ApplicationName>WEB_NET_MANAGEMENT</com:ApplicationName>
  </soapenv:Header>

  <soapenv:Body>
    <dev:StartDeviceTestRequest>
      <dev:DeviceIdentification>device-01</dev:DeviceIdentification>
    </dev:StartDeviceTestRequest>
  </soapenv:Body>
</soapenv:Envelope>

  xmlns:ns2="http://www.opensmartgridplatform.org/schemas/deviceinstallation/2014/10"
>
  <SOAP-ENV:Header/>

  <SOAP-ENV:Body>
    <ns2:StartDeviceTestAsyncResponse>
      <ns3:CorrelationUid>LianderNetManagement||device-01||20160104155530194</ns3:CorrelationUid>
      <ns3:DeviceId>device-01</ns3:DeviceId>
    </ns2:StartDeviceTestAsyncResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

  xmlns:dev="http://www.opensmartgridplatform.org/schemas/deviceinstallation/2014/10"
>
  <soapenv:Header>
    <com:OrganisationIdentification>LianderNetManagement</com:OrganisationIdentification>
    <com:UserName>liander gebruiker</com:UserName>
    <com:ApplicationName>WEB_NET_MANAGEMENT</com:ApplicationName>
  </soapenv:Header>

  <soapenv:Body>
    <dev:StartDeviceTestAsyncRequest>
      <dev:AsyncRequest>
        <com:CorrelationUid>LianderNetManagement||device-01||20160104155530194</com:CorrelationUid>
        <com:DeviceId>device-01</com:DeviceId>
      </dev:AsyncRequest>
    </dev:StartDeviceTestAsyncRequest>
  </soapenv:Body>
</soapenv:Envelope>
```
OSLP StartSelfTestRequest message sent to 'device-01':

```
startSelfTestRequest {
}
```

OSLP StartSelfTestResponse message sent to platform:

```
startSelfTestResponse {
  status: OK
}
```
StopSelfTest

StopSelfTest messages

Description
Request that notifies the device to switch all light relays off.
Response communicates status and the result of the test.

Message definitions

message StopSelfTestRequest {
  optional bool present = 1 [default = true];
}

message StopSelfTestResponse {
  required Status status = 1;
  required bytes selfTestResult = 2; // [(nanopb).max_size = 1];
}

Data types

enum Status {
  OK = 0;
  FAILURE = 1; // general failure
  REJECTED = 2; // request received in wrong state
}

Example

Soap requests and responses sent to and from platform:

<soapenv:Header>
  <com:OrganisationIdentification>LianderNetManagement</com:OrganisationIdentification>
  <com:UserName>liander gebruiker</com:UserName>
  <com:ApplicationName>WEB_NET_MANAGEMENT</com:ApplicationName>
</soapenv:Header>
<soapenv:Body>
  <dev:StopDeviceTestRequest>
    <dev:DeviceIdentification>device-01</dev:DeviceIdentification>
  </dev:StopDeviceTestRequest>
</soapenv:Body>
</soapenv:Envelope>

<SOAP-ENV:Header />
<SOAP-ENV:Body>
  <ns2:StopDeviceTestAsyncResponse xmlns:ns2="http://www.opensmartgridplatform.org/schem
  <ns2:AsyncResponse>
    <ns3:CorrelationUid>LianderNetManagement|||device-01|||20160104160800238</ns3:CorrelationUid>
    <ns3:DeviceId>device-01</ns3:DeviceId>
  </ns2:AsyncResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

<soapenv:Header>
  <com:OrganisationIdentification>LianderNetManagement</com:OrganisationIdentification>
  <com:UserName>liander gebruiker</com:UserName>
  <com:ApplicationName>WEB_NET_MANAGEMENT</com:ApplicationName>
</soapenv:Header>
<soapenv:Body>
  <dev:StopDeviceTestAsyncRequest>
    <dev:AsyncRequest>
      <com:CorrelationUid>LianderNetManagement|||device-01|||20160104160800238</com:CorrelationUid>
      <com:DeviceId>device-01</com:DeviceId>
    </dev:AsyncRequest>
  </dev:StopDeviceTestAsyncRequest>
</dev:StopDeviceTestAsyncRequest>
OSLP StopSelfTestRequest sent to 'device-01':
stopSelfTestRequest {
}

OSLP StopSelfTestResponse sent to platform:
stopSelfTestResponse {
  status: OK
  selfTestResult: "\000"
}
SetLight

SetLight messages

Description

Request that notifies the device to switch on or off one or several light relays, optionally with a dim-value per relay. If optional value 'index' is omitted, all relays configured as light are switched. In that case, all light relays will switch using only 1 LightValue instance for 'values'. In case the value 'index' is included, multiple instances of LightValue can be used (up to 6), each indicating a particular relay. If optional value 'dimValue' is omitted, then default values of 0 and 100 will be assumed for either 'on = false' or 'on = true'.

Response communicates status.

Message definitions

message SetLightRequest {
    repeated LightValue values = 1; // [(nanopb).max_count = 6];
}

message SetLightResponse {
    required Status status = 1;
}

Data types

message LightValue {
    optional bytes index = 1; // [(nanopb).max_size = 1]; // index number of connected light
    required bool on = 2;
    optional bytes dimValue = 3; // [(nanopb).max_size = 1]; // 1 - 100 %
}

enum Status {
    OK = 0;
    FAILURE = 1; // general failure
    REJECTED = 2; // request received in wrong state
}

Example

Soap requests and responses sent to and from platform:

    <SOAP-ENV:Header>
    < OrganisationIdentification xmlns="http://www.opensmartgridplatform.org/schemas/common">
    < ApplicationName xmlns="http://www.opensmartgridplatform.org/schemas/common">WEB_OWNEL</ApplicationName>
    < UserName xmlns="http://www.opensmartgridplatform.org/schemas/common">liander gebruiker</UserName>
    </ OrganisationIdentification>
    < SOAP-ENV:Header/>
    <SOAP-ENV:Body>
    <ns2:SetLightRequest xmlns:ns2="http://www.opensmartgridplatform.org/schemas/publiclig
    <ns2:DeviceIdentification>device-01</ns2:DeviceIdentification>
    <ns2:LightValue>
    <ns2:On>true</ns2:On>
    </ns2:LightValue>
    </ns2:SetLightRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
    <ns2:SetLightAsyncResponse xmlns:ns2="http://www.opensmartgridplatform.org/schemas/pub
    <ns2:AsyncResponse>
    <ns3:CorrelationUid>LianderNetManagement|||device-01|||20160105121022551</ns3:CorrelationUid>
    <ns3:DeviceId>device-01</ns3:DeviceId>
    </ns2:AsyncResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
OSLP SetLightRequest sent to 'device-01':

```
setLightRequest {
    values {
        on: true
    }
}
```

OSLP SetLightResponse sent to platform:

```
setLightResponse {
    status: OK
}
```
SetTransition

SetTransition messages

Description
Request which informs a device of a daylight transition: it has become dark (sunset) or light (sunrise). The device will switch the relays, which have schedule entries for transition messages. The optional 'time' value can be used to indicate a switch time. If the optional 'time' value is omitted the device should switch immediately. See light schedule-entry for more information regarding switch schedules.

Response communicates status.

Message definitions
message SetTransitionRequest {
  required TransitionType transitionType = 1; // Night-Day or Day-Night transition
  optional string time = 2; // [(nanopb).max_size = 7]; // - format hhmms UTC
}
message SetTransitionResponse {
  required Status status = 1;
}

Data types
data TransitionType {
  NIGHT_DAY = 0;
  DAY_NIGHT = 1;
}
data Status {
  OK = 0;
  FAILURE = 1; // general failure
  REJECTED = 2; // request received in wrong state
}

Example
Soap requests and responses sent to and from platform:

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:adh="http:
<soapenv:Header>
  <com:OrganisationIdentification>LianderNetManagement</com:OrganisationIdentification>
  <com:UserName>liander gebruiker</com:UserName>
  <com:ApplicationName>SoapUI</com:ApplicationName>
</soapenv:Header>
<soapenv:Body>
  <adh:SetTransitionRequest>
    <adh:DeviceIdentification>device-01</adh:DeviceIdentification>
    <adh:TransitionType>NIGHT_DAY</adh:TransitionType>
    <adh:Time>07:55:01</adh:Time>
  </adh:SetTransitionRequest>
</soapenv:Body>
</soapenv:Envelope>

<SOAP-ENV:Header/>
<SOAP-ENV:Body>
  <ns2:SetTransitionAsyncResponse xmlns:ns2="http://www.opensmartgridplatform.org/schema
    <ns3:CorrelationUid>LianderNetManagement|||device-01|||20160106155501582</ns3:CorrelationUid>
    <ns3:DeviceId>device-01</ns3:DeviceId>
  </ns2:AsyncResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
OSLP SetTransitionRequest sent to 'device-01':

```
setTransitionRequest {
  transitionType: NIGHT_DAY
  time: "075501"
}
```

OSLP SetTransitionResponse sent to platform:

```
setTransitionResponse {
  status: OK
}
```
GetStatus

GetStatus messages

Description

Request that requires the device to send the status of all relays, current network link and preferred network link, the type of configuration (PSLD vs SSD), and the event notification mask which has been set. Further, many optional values can be set by the device, like serial number, MAC address, memory sizes, current firmware version, current IP address, etc.

Response which confirms the GetStatusRequest has been executed and returns the current status for all of the relays and other information or rejects the GetStatusRequest.

Message definitions

message GetStatusRequest {
  optional bool present = 1 [default = true];
}

message GetStatusResponse {
  required Status status = 1;
  repeated LightValue value = 2; [(nanopb).max_count = 6];
  required LinkType preferredLinktype = 3;
  required LinkType actualLinktype = 4;
  required LightType lightType = 5;
  required uint32 eventNotificationMask = 6; // Bitmask for max 32 events, using NotificationBit for bit positions.
  optional uint32 numberOfOutputs = 7; // Hardware - The number of outputs of this device.
  optional uint32 dcOutputVoltageMaximum = 8; // Hardware - DC output voltage maximum.
  optional uint32 dcOutputVoltageCurrent = 9; // Hardware - DC output current.
  optional uint32 maximumOutputPowerOnDcOutput = 10; // Hardware - Maximum output power.
  optional bytes serialNumber = 11; // [(nanopb).max_size = 18]; // Hardware - Serial number of this device.
  optional bytes macAddress = 12; // [(nanopb).max_size = 6]; // Hardware - MAC-address of this device.
  optional string hardwareId = 13; // [(nanopb).min_size = 10, (nanopb).max_size = 25]; // Hardware - The hardware ID of this device.
  optional uint32 internalFlashMemSize = 14; // Hardware - The internal flash memory size.
  optional uint32 externalFlashMemSize = 15; // Hardware - The external flash memory size.
  optional uint32 lastInternalTestResultCode = 16; // Hardware - The last internal test result code.
  optional uint32 startupCounter = 17; // Hardware - The startup counter.
  optional string bootLoaderVersion = 18; // Software - The boot loader version.
  optional string firmwareVersion = 19; // Software - The firmware version.
  optional bytes currentConfigurationBackUsed = 20; // [(nanopb).max_size = 6]; // Software - The current configuration bank in use.
  optional string name = 21; // Device - The name of this device.
  optional string currentTime = 22; // Device - Not UTC, the time used in timing operations (adjusted "offset" + summer timing).
  optional string currentIp = 23; // Device - The current IP address of this device.
}

Datatypes

enum Status {
  OK = 0;
  FAILURE = 1; // general failure
  REJECTED = 2; // request received in wrong state
}

message LightValue {
  optional bytes index = 1; // [(nanopb).max_size = 1]; // index number of connected light
  optional bool on = 2; // required bool on 2;
  optional bytes dimValue = 3; // [(nanopb).max_size = 1]; // 1 - 100 %
}

enum LinkType {
  LINK_NOT_SET = 0;
  GPRS = 1;
  CDMA = 2;
  ETHERNET = 3;
}

enum LightType {
  LT_NOT_SET = 0;
  RELAY = 1;
  ONE_TO_TEN_VOLT = 2;
  ONE_TO_TEN_VOLT_REVERSE = 3;
}
DALI = 4;
}

Example

Soap requests and responses sent to and from platform:

```xml
    <SOAP-ENV:Header>
        <OrganisationIdentification xmlns="http://www.opensmartgridplatform.org/schemas/common">
            <ApplicationName xmlns="http://www.opensmartgridplatform.org/schemas/common">SoapUI</ApplicationName>
            <UserName xmlns="http://www.opensmartgridplatform.org/schemas/common">Kevin</UserName>
        </OrganisationIdentification>
    </SOAP-ENV:Header>
    <ns2:GetStatusRequest xmlns:ns2="http://www.opensmartgridplatform.org/schemas/publiclighting/adhocmanagement/2014/10">
        <ns2:DeviceIdentification>device-01</ns2:DeviceIdentification>
    </ns2:GetStatusRequest>
</SOAP-ENV:Envelope>

    <SOAP-ENV:Header>
        <OrganisationIdentification xmlns="http://www.opensmartgridplatform.org/schemas/common">
            <ApplicationName xmlns="http://www.opensmartgridplatform.org/schemas/common">SoapUI</ApplicationName>
            <UserName xmlns="http://www.opensmartgridplatform.org/schemas/common">Kevin</UserName>
        </OrganisationIdentification>
    </SOAP-ENV:Header>
    <ns2:GetStatusAsyncRequest xmlns:ns2="http://www.opensmartgridplatform.org/schemas/publiclighting/adhocmanagement/2014/10">
        <ns2:AsyncRequest>
            <ns3:CorrelationUid>LianderNetManagement|||device-01|||20160106133844686</ns3:CorrelationUid>
            <ns3:DeviceId>device-01</ns3:DeviceId>
        </ns2:AsyncRequest>
    </ns2:GetStatusAsyncRequest>
</SOAP-ENV:Envelope>

    <SOAP-ENV:Header>
        <OrganisationIdentification xmlns="http://www.opensmartgridplatform.org/schemas/common">
            <ApplicationName xmlns="http://www.opensmartgridplatform.org/schemas/common">SoapUI</ApplicationName>
            <UserName xmlns="http://www.opensmartgridplatform.org/schemas/common">Kevin</UserName>
        </OrganisationIdentification>
    </SOAP-ENV:Header>
    <ns2:GetStatusAsyncResponse xmlns:ns2="http://www.opensmartgridplatform.org/schemas/publiclighting/adhocmanagement/2014/10">
        <ns2:AsyncResponse>
            <ns3:CorrelationUid>LianderNetManagement|||device-01|||20160106133844686</ns3:CorrelationUid>
            <ns3:DeviceId>device-01</ns3:DeviceId>
        </ns2:AsyncResponse>
    </ns2:GetStatusAsyncResponse>
</SOAP-ENV:Envelope>

    <SOAP-ENV:Header>
        <OrganisationIdentification xmlns="http://www.opensmartgridplatform.org/schemas/common">
            <ApplicationName xmlns="http://www.opensmartgridplatform.org/schemas/common">SoapUI</ApplicationName>
            <UserName xmlns="http://www.opensmartgridplatform.org/schemas/common">Kevin</UserName>
        </OrganisationIdentification>
    </SOAP-ENV:Header>
    <ns3:GetStatusResponse xmlns:ns3="http://www.opensmartgridplatform.org/schemas/common/2014/10">
        <ns3:Result>OK</ns3:Result>
        <ns3:DeviceStatus>
            <ns3:LightValues>
                <ns3:Index>1</ns3:Index>
                <ns3:On>false</ns3:On>
            </ns3:LightValues>
            <ns3:LightValues>
                <ns3:Index>2</ns3:Index>
                <ns3:On>false</ns3:On>
            </ns3:LightValues>
            <ns3:TariffValues>
                <ns3:Index>3</ns3:Index>
                <ns3:High>true</ns3:High>
            </ns3:TariffValues>
            <ns3:PreferredLinkType>ETHERNET</ns3:PreferredLinkType>
            <ns3:ActualLinkType>ETHERNET</ns3:ActualLinkType>
            <ns3:LightType>RELAY</ns3:LightType>
            <ns3:EventNotifications>DIAG_EVENTS</ns3:EventNotifications>
            <ns3:EventNotifications>HARDWARE_FAILURE</ns3:EventNotifications>
            <ns3:EventNotifications>LIGHT_EVENTS</ns3:EventNotifications>
        </ns3:DeviceStatus>
    </ns3:GetStatusResponse>
</SOAP-ENV:Envelope>
```
OSLP GetStatusRequest sent to 'device-01':

getStatusRequest {
}

OSLP GetStatusResponse sent to platform:

getStatusResponse {
  status: OK
  value {
    index: "\001"
    on: false
  }
  value {
    index: "\002"
    on: false
  }
  value {
    index: "\003"
    on: false
  }
  value {
    index: "\004"
    on: false
  }
  preferredLinktype: ETHERNET
  actualLinktype: ETHERNET
  lightType: RELAY
  eventNotificationMask: 255
  numberOfOutputs: 4
  dcOutputVoltageMaximum: 24000
  dcOutputVoltageCurrent: 0
  maximumOutputPowerOnDcOutput: 15000
  serialNumber: "123456789123456789"
  macAddress: "\330\2009F\027N"
  hardwareId: "SB10"
  internalFlashMemSize: 1048576
  externalFlashMemSize: 8388608
  lastInternalTestResultCode: 0
  startupCounter: 2
  bootloaderVersion: "v1.0"
  firmwareVersion: "W0311g"
  currentConfigurationBackUsed: "\000"
  name: "device-01"
  currentTime: "20160313141247"
  currentIp: "192.168.178.16"
}

OSLP v0.6.1 512
UpdateDeviceSslCertification

UpdateDeviceSslCertification messages

Description
Request to download a new SSL certificate from the certificate server. The device will be given the domain name and URL where the certificate is located.

Message definitions
message UpdateDeviceSslCertificationRequest {
  required string certificateDomain = 1; // [(nanopb).max_size = 100]; // The domain name
  required string certificateUrl = 2; // [(nanopb).max_size = 255]; // The relative path
}
message UpdateDeviceSslCertificationResponse {
  required Status status = 1;
}

Datatypes
enum Status {
  OK = 0;
  FAILURE = 1; // general failure
  REJECTED = 2; // request received in wrong state
}

Example
Soap requests and responses sent to and from platform:

```xml
  xmlns:ns1="http://www.opensmartgridplatform.org/schemas/devicemanagement/2014/10"
  xmlns:ns2="http://www.opensmartgridplatform.org/schemas/protocol/2014/10">
  <soapenv:Header>
    <ns:ApplicationName>SoapUI</ns:ApplicationName>
    <ns:UserName>Kevin</ns:UserName>
    <ns:OrganisationIdentification>LianderNetManagement</ns:OrganisationIdentification>
  </soapenv:Header>
  <ns1:UpdateDeviceSslCertificationRequest>
    <!--type: Identification-->device-01</ns1:DeviceIdentification>
    <ns1:Certification>
      <ns1:certificateDomain>cert-server</ns1:certificateDomain>
      <ns1:certificateUrl>/certs/new-cert.pem</ns1:certificateUrl>
    </ns1:Certification>
  </ns1:UpdateDeviceSslCertificationRequest>
</soapenv:Body>
</SOAP-ENV:Envelope>
```

```xml
  xmlns:ns3="http://www.opensmartgridplatform.org/schemas/protocol/2014/10">
  <SOAP-ENV:Header>
    <ns:ApplicationName>SoapUI</ns:ApplicationName>
    <ns:UserName>Kevin</ns:UserName>
    <ns:OrganisationIdentification>LianderNetManagement</ns:OrganisationIdentification>
  </SOAP-ENV:Header>
  <ns2:UpdateDeviceSslCertificationAsyncResponse xmlns:ns2="http://www.opensmartgridplatform.org/schemas/protocol/2014/10">
    <ns3:CorrelationUid>LianderNetManagement|||device-01|||20160305115500062</ns3:CorrelationUid>
    <ns3:DeviceId>device-01</ns3:DeviceId>
  </ns2:UpdateDeviceSslCertificationAsyncResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```xml
  xmlns:ns1="http://www.opensmartgridplatform.org/schemas/devicemanagement/2014/10">
  <soapenv:Header>
    <ns:ApplicationName>SoapUI</ns:ApplicationName>
    <ns:UserName>Kevin</ns:UserName>
    <ns:OrganisationIdentification>LianderNetManagement</ns:OrganisationIdentification>
  </soapenv:Header>
  <ns1:UpdateDeviceSslCertificationAsyncRequest>
    <ns1:AsyncRequest>
      <!--type: Identification-->device-01</ns1:DeviceIdentification>
    </ns1:AsyncRequest>
  </ns1:UpdateDeviceSslCertificationAsyncRequest>
</soapenv:Body>
</soapenv:Envelope>
```
<ns:CorrelationUid>
<ns:CorrelationUid>LianderNetManagement|||device-01|||20160305115500062</ns:CorrelationUid>
</ns:CorrelationUid>

<ns:DeviceId>device-01</ns:DeviceId>
</ns1:AsyncRequest>
</soapenv:Body>
</soapenv:Envelope>

<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<ns2:UpdateDeviceSslCertificationResponse xmlns:ns2="http://www.opensmartgridplatform.org/schemas/devicemanagement/2014/10">
<ns2:Result>OK</ns2:Result>
</ns2:UpdateDeviceSslCertificationResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

OSLP messages:
updateDeviceSslCertificationRequest {
  certificateDomain: "cert-server"
  certificateUrl: "/certs/new-cert.pem"
}

updateDeviceSslCertificationResponse {
  status: OK
}
SetDeviceVerificationKey

SetDeviceVerificationKey messages

Description

Request to switch to a new Platform public key used for verifying OSLP envelopes by the device. The base-64 encoded version of the key will be sent to the device, which is equivalent to the content of a PEM file (only the certificate chunk, not the headers).

Message definitions

message SetDeviceVerificationKeyRequest {  
  required bytes certificateChunk = 1; // [(nanopb).max_size = 138]; // Verification key /
}

message SetDeviceVerificationKeyResponse {  
  required Status status = 1;
}

Datatypes

datatype Status {  
  OK = 0;  
  FAILURE = 1; // general failure  
  REJECTED = 2; // request received in wrong state
}

Example

Soap requests and responses sent to and from platform:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
 xmlns:ns="http://">
  <soapenv:Header>
    <ns:ApplicationName>SoapUI</ns:ApplicationName>
    <ns:UserName>Kevin</ns:UserName>
    <ns:OrganisationIdentification>LianderNetManagement</ns:OrganisationIdentification>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:SetDeviceVerificationKeyRequest>
      <!--type: Identification-->  
      <ns1:DeviceIdentification>device-01</ns1:DeviceIdentification>
      <!--type: VerificationKey-->  
      <ns1:VerificationKey>MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEow7CWR7EiNDRT1XQ/h1UrLE2</ns1:VerificationKey>
    </ns1:SetDeviceVerificationKeyRequest>
  </soapenv:Body>
</soapenv:Envelope>

 xmlns:
  <ns2:SetDeviceVerificationKeyAsyncResponse xmlns:ns3="http://www.opensmartgridplatform
  <ns3:CorrelationUid>LianderNetManagement|||device-01|||20160305122132785</ns3:CorrelationUid>
  <ns3:DeviceId>device-01</ns3:DeviceId>
</ns2:SetDeviceVerificationKeyAsyncResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
 xmlns:ns="http://">
  <soapenv:Header>
    <ns:ApplicationName>SoapUI</ns:ApplicationName>
    <ns:UserName>Kevin</ns:UserName>
    <ns:OrganisationIdentification>LianderNetManagement</ns:OrganisationIdentification>
  </soapenv:Header>
  <soapenv:Body>
    <ns1:SetDeviceVerificationKeyAsyncRequest>
      <!--type: CorrelationUid-->  
      <ns:CorrelationUid>LianderNetManagement|||device-01|||20160305122132785</ns:CorrelationUid>
</soapenv:Body>
</soapenv:Envelope>
```
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:SetDeviceVerificationKeyResponse xmlns:ns2="http://www.opensmartgridplatform.org/schemas/devicemanagement/2014/10">
      <ns2:Result>OK</ns2:Result>
    </ns2:SetDeviceVerificationKeyResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

OSLP messages:

setDeviceVerificationKeyRequest { 
  certificateChunk: "MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEow7CWR7EiNDرت1XQ/h1UrLE24y3BkA58; 
}

setDeviceVerificationKeyResponse { 
  status: OK
}
SwitchFirmware

SwitchFirmware messages

Description
Request to switch from the current firmware version to the other firmware version, indicated by the argument newFirmwareVersion.

Message definitions
message SwitchFirmwareRequest {
    required string newFirmwareVersion = 1; // [(nanopb).max_size = 6]; // The version of the firmware which should be installed.
}
message SwitchFirmwareResponse {
    required Status status = 1; // FIRMWARE_EVENTS_ACTIVATING Event will be sent, after the firmware change has completed.
}

Datatypes
enum Status {
    OK = 0;
    FAILURE = 1; // general failure
    REJECTED = 2; // request received in wrong state
}

Example
Soap requests and responses sent to and from platform:

    xmlns:ns1="http://www.opensmartgridplatform.org/schemas/common/firmwaremanagement/2014/10"
    xmlns:ns3="http://www.opensmartgridplatform.org/schemas/common/2014/10"
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:SOAP-ENV:Body">
    <ns1:SwitchFirmwareRequest
        <!--type: Identification-->>
    </ns1:SwitchFirmwareRequest>
</soapenv:Body>
</soapenv:Envelope>

    xmlns:ns1="http://www.opensmartgridplatform.org/schemas/common/firmwaremanagement/2014/10"
    xmlns:ns3="http://www.opensmartgridplatform.org/schemas/common/2014/10"
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:SOAP-ENV:Body">
    <ns2:SwitchFirmwareAsyncRequest xmlns:ns2="http://www.opensmartgridplatform.org/schemas/common/firmwaremanagement/2014/10"
        xmlns:ns3="http://www.opensmartgridplatform.org/schemas/common/2014/10"
        xmlns:ns4="http://www.opensmartgridplatform.org/schemas/common/2014/10"
        xmlns:ns5="http://www.opensmartgridplatform.org/schemas/common/2014/10"
        xmlns:ns6="http://www.opensmartgridplatform.org/schemas/common/2014/10"
        xmlns:ns7="http://www.opensmartgridplatform.org/schemas/common/2014/10"
        xmlns:ns8="http://www.opensmartgridplatform.org/schemas/common/2014/10"
        xmlns:ns9="http://www.opensmartgridplatform.org/schemas/common/2014/10"
        xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
        xmlns:SOAP-ENV:Body">
    </ns2:SwitchFirmwareAsyncRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

    xmlns:ns1="http://www.opensmartgridplatform.org/schemas/common/firmwaremanagement/2014/10"
    xmlns:ns3="http://www.opensmartgridplatform.org/schemas/common/2014/10"
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:SOAP-ENV:Body">
    <ns1:SwitchFirmwareAsyncRequest
        <!--type: CorrelationUid-->
    </ns1:SwitchFirmwareAsyncRequest>
</soapenv:Body>
</soapenv:Envelope>
<ns:DeviceId>device-01</ns:DeviceId>
</ns1:AsyncRequest>
</ns1:SwitchFirmwareAsyncRequest>
</soapenv:Body>
</SOAP-ENV:Envelope>

<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<ns2:SwitchFirmwareResponse xmlns:ns2="http://www.opensmartgridplatform.org/schemas/common/firmwaremanagement/2014/10">
<ns2:Result>OK</ns2:Result>
</ns2:SwitchFirmwareResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

**OSLP messages:**

```
switchFirmwareRequest {
    newFirmwareVersion: "W0311g"
}

switchFirmwareResponse {
    status: OK
}
```
SwitchConfiguration

SwitchConfiguration messages

Description
Request to switch from the current (active) configuration set to the other configuration set, indicated by the configuration set index.

Message definitions
message SwitchConfigurationRequest {
  required bytes newConfigurationSet = 1; // [(nanopb).max_count = 1]; // The index of the
}
message SwitchConfigurationResponse {
  required Status status = 1; // FIRMWARE_EVENTS_CONFIGURATION_CHANGED Event will be sent,
}

Datatypes
enum Status {
  OK = 0;
  FAILURE = 1; // general failure
  REJECTED = 2; // request received in wrong state
}

Example
Soap requests and responses sent to and from platform:

```xml
<soapenv:Header>
  <ns:ApplicationName>SoapUI</ns:ApplicationName>
  <ns:UserName>Kevin</ns:UserName>
  <ns:OrganisationIdentification>LianderNetManagement</ns:OrganisationIdentification>
</soapenv:Header>
<soapenv:Body>
  <ns1:SwitchConfigurationRequest>
    <!--type: Identification-->  
    <ns1:DeviceIdentification>device-01</ns1:DeviceIdentification>

```
<ns:DeviceId>device-01</ns:DeviceId>
</ns1:AsyncRequest>
</ns1:SwitchConfigurationAsyncRequest>
</soapenv:Body>
</soapenv:Envelope>

 <SOAP-ENV:Header/>
 <SOAP-ENV:Body>
  <ns2:SwitchConfigurationResponse xmlns:ns2="http://www.opensmartgridplatform.org/schemas/configurationmanagement/2014/10">
   <ns2:Result>OK</ns2:Result>
  </ns2:SwitchConfigurationResponse>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

// OSLP messages:

switchConfigurationRequest {
  newConfigurationSet: "1"
}

switchConfigurationResponse {
  status: OK
}
MQTT Documentation

MQ Telemetry Transport Protocol

MQTT is a lightweight publish/subscribe messaging protocol designed for constrained devices and networks.

Dependencies

The MQTT Protocol Adapter uses the HiveMQ MQTT client `com.hivemq:hivemq-mqtt-client` which supports MQTT 3. The MQTT Protocol Simulator also uses HiveMQ, as well as the Moquette MQTT broker `io.moquette:moquette-broker`. The reason for using Moquette is that there is no Maven dependency for the HiveMQ broker component.

Simulator

The simulator runs an MQTT broker and a client which repeatedly publishes messages as specified in its `mqtt_simulator_spec.json` file.

Example:

```json
{
  "brokerHost": "0.0.0.0",
  "brokerPort": 8883,
  "startupPauseMillis": 5000,
  "messages": [
    {
      "topic": "TST-01/measurement",
      "payload": "TST-01; 220.1; 220.2; 220.3; 5.1; 5.2; 5.3; 7.1; 7.2; 7.3;",
      "pauseMillis": 30000
    },
    {
      "topic": "TST-01/congestion",
      "payload": "TST-01; 5.1; 5.2; 5.3; 7.1; 7.2; 7.3;",
      "pauseMillis": 30000
    }
  ]
}
```

The spec file above will pause for 5 seconds and then start publishing a measurement and congestion message with intervals of 30 seconds. The message payload is assumed to be a String and not validated or parsed in any way.

The protocol adapter will establish a connection with the simulator and subscribe to the default topics `+/measurement`, `+/congestion`. This means measurement and congestion messages from any device (`+`).

A configurable default QoS value is used (See `com.hivemq.client.mqtt.datatypes.MqttQos` for the values).

If not yet present, an MqttDevice is saved in the database of the protocol adapter with the values used. If the MqttDevice is updated in the database, the updated values will be used for subsequent communication. There is not yet any means to update this data, other than manual updating in the database.
Support

CHAPTER 6: Support

There are multiple options for support

Community support

Community members can help you on voluntary basis. See the open source and community section for more information where you can ask your questions.

Commercial support

Currently there's no commercial support available. If you would like to provide commercial support, contact us and we'll add your company name here.
License

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NONINFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
**Glossary**

**COSEM**

COmpanion Specification for Energy Metering, defines a set of objects to exchange data with Smart Meters using DLMS protocol.

- 1.5. Platform components description
- 1.2. Architecture introduction
- 1.9.5. Technology Stack
- 4.6.3. Use cases
- 4.6.1.6.27. ConfigureDefinableLoadProfile
- 5.2.1. DLMS device simulator
- 5.2. DLMS / COSEM

**DLMS**

Device Language Message Specification, the protocol used to communicate with Smart Meters and other smart grid devices.

- 1.7. Logical Authorisation Model
- 1.5. Platform components description
- 1.2. Architecture introduction
- 1.9.4. Protocol Layer
- 1.9.5. Technology Stack
- 4.6. SmartMetering
- 4.6.3. Use cases
- 4.6.1.9.9. GetProfileGenericData
- 4.6.1.6.29. SetMbusUserKeyByChannel
- 5.2.1. DLMS device simulator
- 5.2. DLMS / COSEM
- 5.3. OSLP
- 5. Protocols
- 2.2.1. Add a device

**DSMR**

Dutch Smart Meter Requirements, a set of rules that describe how to use Smart Meters using DLMS/COSEM, defined by Dutch grid operators. For more information [click here](http://www.netbeheernederland.nl/themas/dossier/documenten/?pageindex=7)

- 4.6.1.8.1. FindEvents
- 4.6.1.6.25. GetConfigurationObject
- 4.6.1.6.15. SetConfigurationObject
- 5.2. DLMS / COSEM

**DTO Object**

Data Transfer Object. See [Wikipedia](https://en.wikipedia.org/wiki/Data_transfer_object)

- 1.9. Technical Overview

**Domain driven design**

See [Wikipedia](https://en.wikipedia.org/wiki/Domain-driven_design)

- 3.2. Developers 101
- 1.4. Architecture Principles

**GXF**

Grid eXchange Fabric, formerly known as the Open Smart Grid Platform (OSGP). GXF is an open, generic, scalable and independent 'Internet of Things' platform, which enables various connected smart objects in the public space to be easily controlled and monitored. The GXF project is built using open source tools and standards.

- 3.7. Code of Conduct
- 3.5. Communication and Contact
IEC 61850

IEC 61850 is an international standard defining communication protocols for intelligent electronic smart devices at electrical substations. See [IEC](https://www.iec.ch/smartgrid/standards/)

- 1.5. Platform components description
- 1.9.4. Protocol Layer
- 1.9.5. Technology Stack
- 4.4. Microgrids
- 4.2.2. Light Schedules
- 4.7. Guidelines to add a new domain to GXF
- 5.1.2.3. EventNotification
- 5.1.2.1. FlexOVL 540_171101_2_out.icd
- 5.1.2.2. GetStatus
- 5.1.2. FlexOVL 540_171101_2
- 5. IEC61850
- 5.1.1.13. SetLight
- 5.1.1.6. EventNotification
- 5.1.1.8. GetFirmwareVersion
- 5.1.1.5. GetStatus
- 5.1.1.2. RegisterDevice
- 5.1.1.4. SetConfiguration
- 5.1.1.5. SetEventNotifications
- 5.1.1.3. GetConfiguration
- 5.1.1.10. SetReboot
- 5.1.1.7. SetSchedule
- 5.1.1.4. SetTransition
- 5.1.1.11. StartSelfTest
- 5.1.1.12. StopSelfTest
- 5.1.1.16. UpdateDeviceSslCertification
- 5.1.1.9. UpdateFirmware
- 5.1.1. SWDevice-010805
- 5. Protocols

LMD

Light Measurement Device, a smart grid device which uses up to 4 light sensor arrays to determine light or dark for several areas.

- 5.1.2. FlexOVL 540_171101_2

MQTT

MQTT stands for MQ Telemetry Transport. It is a publish/subscribe, extremely simple and lightweight messaging protocol, designed for constrained devices and low-bandwidth, high-latency or unreliable networks. See [MQTT](http://mqtt.org/)

- 5.4. MQTT
- 5. Protocols

OSGP

Open Smart Grid Platform, see GXF.
OSLP

Open Street Light Protocol, the protocol used to communicate with SSLD and other smart grid devices.

- 3.2. Developers 101
- 1.3. Architecture functional layers
- 1.8.2. Internationalization and localization
- 1.7. Logical Authorization Model
- 1.8.6. Performance
- 1.5. Platform components description
- 1.9.4. Protocol Layer
- 1.9. Technical Overview
- 1.9.5. Technology Stack
- 1.8.1. TimeBehavior
- 1.8.3. Security
- 4.2.2. Light Schedules
- 5.1. IEC61850
- 5.3. OSLP
- 5.3.1. OSLP v0.5.1
- 5.3.2. OSLP v0.6.1
- 5.3.1.1. Protobuf Contract
- 5.3.2.12. SetReboot
- 5.3.2.13. ConfirmRegisterDevice
- 5.3.2.4. GetConfiguration
- 5.3.2.10. GetFirmwareVersion
- 5.3.2.17. GetStatus
- 5.3.2.20. RegisterDevice
- 5.3.2.18. ResumeSchedule
- 5.3.2.5. SetConfiguration
- 5.3.2.19. SetDeviceVerificationKey
- 5.3.2.6. SetLight
- 5.3.2.15. SetEventNotifications
- 5.3.2.7. SetSchedule
- 5.3.2.16. SetTransition
- 5.3.2.13. StartSelfTest
- 5.3.2.14. StopSelfTest
- 5.3.2.21. StopConfiguration
- 5.3.2.20. SwitchConfiguration
- 5.3.2.18. UpdateDeviceSslCertification
- 5.3.2.11. UpdateFirmware
- 5.3.2.1. Protobuf Contract
- 5. Protocols
- 2.2.1. Add a device
- 2.5. FAQ
- 2.2.1.1. Using the Demo App
- 2.1.3.1. Using SoapUi

Glossary

Grid eXchange Fabric Documentation
PSLD

Public Street Lighting Device, a smart grid device that is used to control and monitor a single street light.

- 1.3. Architecture functional layers
- 5.3.2. OSLP v0.6.1
- 5.3.1.1. Protobuf Contract
- 5.3.2.4. GetConfiguration
- 5.3.2.17. GetStatus
- 5.3.2.1. Protobuf Contract

SOAP Webservice

The open smart grid platform offers a Spring Framework SOAP Webservice.

SSLD

Sub Station Lighting Device, a smart grid device that is used to control and monitor public lighting (several street lights) and tariff switching for an area.