ISY
Energy Management System
Developer's Manual

Web Services SDK and REST Interface
Based on firmware 3.3.10

TABLE OF CONTENTS

0.0 Revision History

Date/Firmware	Type	Change	Description
2012/10/12 **3.3.4**	REST	NEW	-Metering log
2012/05/15 **3.3.1**	WSDL	Modify	-Moved Zigbee to zigbee.xsd -Moved SEP objects to sepobjs.xsd -Updated documentation for metering and changes as above -Reset/clear events -Support for Querying: --Last Message --Current Price --Scheduled Prices --Scheduled DRLC Events
2012/03/26 **3.2.2**	WSDL DOC	Modify	All SEP Events now have datetime represented in both NTP as well as readable string
2012/03/16 **3.2.0**	DOC	Naming	OR to ISY
2011/12/13 **3.1.15**	DOC	FIX	AMI Event Control is _12 and not _17
2011/06/09 – **3.1.4**	WSDL	NEW	Added ProviderId to UserElectricityOptions

1. Introduction

ISY is an energy management system platform based on ISY framework. As such, all ISY interfaces, services, and events are applicable to ISY as well.

ISY adds support for energy management use cases as well as native support for Smart Grid standards and interfaces such as SEP (Smart Energy Profile).

SEP Profile is a set of instructions which can be sent by the utilities to either instruct the HAN (Home Area Network) to orchestrate certain scenarios to achieve load shedding/demand reduction or provides useful information to the consumer so that they could make informed decisions with respect to their energy utilization.

ISY can natively process and orchestrate the HAN based on SEP events/instructions. All SEP events have a Start Time and duration. Duration could be forever – or till a Stop event has been sent.

Although detailed discussion on SEP is outside the scope of this document, but understanding the following four instructions, as defined by SEP, are essential for developing applications for ISY:

1.1 DRLC (Demand Response Load Control)

DRLC is an instruction sent by the utility, or a DR service provider, to cause the HAN to change its operational characteristics based on the following set of attributes:

Criticality – the level of importance for this event

Duty Cycle – the percent of time in which a device must be cycled on/off. Usually applied to load controllers such as pool pumps and water heaters

Average Load Adjustment – the percentage by which adjustable loads should shed their loads. Usually applied to dimmers and variable load devices

Heating Setpoint – precise heating setpoint to which thermostats must be adjusted to. This is mutually exclusive of Heating Setpoint Offset

Cooling Setpoint – precise cooling setpoint to which thermostats must be adjusted to. This is mutually exclusive of Cooling Setpoint Offset

Heating Setpoint Offset – The offset by which the Heating Setpoint for thermostats must be decreased

Cooling Setpoint Offset – The offset by which the Cooling Setpoint for thermostats must be increased

Device Class – This attribute defines the classes of devices (such as pool pumps) that this event must be applied to.

Note that all DRLC events must be explicitly opted into by the customer.

1.2 Message

Message is an informational signal sent by the utility to notify the customer of some event. Messages come in two types:

i. Those requiring customer to confirm receipt
ii. Those that do not require the customer to confirm receipt

1.3 Price

Price is an informational signal sent by the utility to notify the customer of pricing information per tier. CPP (Critical Peak Price) is sent on Tier 5.

1.4 Meter

Meter is the information coming directly from an AMI/Smart Grid meter using SEP protocol. Meter provides such information as current consumption per tier or even current production (i.e. Solar Panels).

Theoretically, all SEP enabled Meters would be able to provide all the above messages/signals. ISY, when equipped with Zigbee SEP module, can natively communicate with the meter and process the information and take actions based on user defined use cases.

This said, however, since not all meters are yet equipped with Zigbee SEP modules, UDI has come up with Web Services version of the first 3 signals (DRLC, Message, and Price) and thus allowing for the same signals to be sent using Web Services. In UDI Vernacular, we call these Web Services Broadband SEP.

Broadband SEP is an optional module which might need to be activated. If you do not have this module enabled, please contact sales@universal-devices.com with your UUID (Help | About) and request for this module to be activated.

The beauty of Broadband SEP is that you can use 3rd party tools, such as SOAP UI to consume SEP Services provided by ISY and generate or modify events at will.

As mentioned before, ISY is a full-fledged energy management system. In order to provide the greatest level of compatibility and flexibility, ISY supports SEP signal from either Web Services or directly from Zigbee meters.

ISY has the capability of receiving, scheduling, and running up to 10 events for each type of DRLC, Price, and Message events. Furthermore, all events are persisted in ISY's file system so that, in case of power failure, they can be restored and restarted/rescheduled.

In addition to full ISY Framework's programmability options available to the users, ISY provides the developer with simple configuration files which can be used in a majority of cases to make customer setup easier and more user friendly. For more information on configuration options for SEP events, please see section 3.

2. Getting Started

ISY is based on the same framework as ISY and therefore communications and event infrastructure follow the same paradigm. If you have not yet reviewed ISY's WSDK Developer's guide, please send an email to sales@universal-devices.com.

If you do not already have Broadband SEP (BSEP) module installed on ISY, please send an email to sales@universal-devices.com with your UUID (Help | About) and your desire to have Broadband SEP module activated.

Once you are successfully communicating with ISY and have BSEP installed, then:

1. Go to http://isy:port/desc (or http://your.isy.ip.address:port/desc)

   You will be presented with the description of services provided by ISY. In the <serviceList> element, look for **UDISEPWebServices** as the <serviceType>. What you are looking for is the URL for Web services binding. This URL is defined <SCPDURL> (see below):

```
<serviceList>
      <service>
            Default ISY Framework service descriptions
      </service>
      <service>
            <serviceType>UDISEPWebServices</serviceType>
            <serviceId>
                  uuid:00:03:f4:03:65:96-UDISEPWebServices
            </serviceId>
            <SCPDURL>/sepServices.wsdl</SCPDURL>
            <controlURL>/sepServices</controlURL>
      </service>
</serviceList>
```

2. Now, all you need to do is point your SOAP client to:
   http://your.isy.ip.address:port/[value for SCPDURL] and import BSEP web services

3. Get ISY Configuration (GetISYConfig Web Service or /rest/config) and ensure that Broadband SEP module is installed (id = 21080). For more information on Configuration Resources and Modules, please consult ISY WSDK Developer's Guide
4. All BSEP web services are defined in the WSDL returned by step 2. These include all SEP event objects and configuration definitions
5. If you are using SOAP UI, you can immediately communicate and issue SEP events to ISY

Please note that all requests require the Authorization header. Furthermore, if you wish to receive SEP events, please do make sure that you have subscribed to ISY (please consult WSDK Developer's Guide).

2.1 SEP Event Lifecycle

All SEP Events go through a specific life cycle the starting point of which is Expired or Done. These states are defined by the **eventState** attribute of all events and enumerated by **SEPEventState** both of which are defined in the WSDL.

All SEP Events have a Start Time (defined by the **startTime** element) and Duration (defined by **duration** element in minutes). Two special cases:

a. If Start Time is 0 or not provided, ISY considers the event's Start Time to be **now**
b. If Duration is 0, not provided, or 65535 (max UINT2), ISY considers the duration to be forever or till a Stop event for the same type has been issued

When events are received by ISY, the following checks are made to make sure the event is valid:

a. Whether or not it has the correct Enrollment Group (defined by **enrollmentGroup** element on all events) matching that already configured via the **EnrollmentGroup** element in **UserElectricityOptions** configuration (see section 3). If they do not match, the event is discarded as invalid and logged
b. Whether or not the Event ID (defined by the **eventId** element) is not already being used by other events in the system. If another event with the same Event ID is found, the event is discarded as invalid and logged
c. Whether or not the event is in the past. If so, event is discarded as invalid and logged
d. Whether or not there are any conflicts with existing events in the system
    i. With the exception of **Price** event, there can only be one event for any specific duration in time. If there are any conflicts, the event is considered to have a schedule conflict, discarded as invalid and logged
    ii. In the case of **Price** event, multiple events could be running in the same time interval as long as they have different Price Tiers (defined by **tier** element in **sepobjs.SEPPriceObject**

If the event is considered valid, then

a. It will go to the **Running** state if it should be started
b. It will go to the **Scheduled** state if it should be scheduled

When an event goes into **Running** state, ISY checks the User options for that event to see if any actions should be taken and, if so, it executes on those actions.

Events are periodically checked to see whether or not they should be expired in which case they will go to the **Done** state.

From a BSEP perspective, events are started using the **SEPStart[Event]** service and can be stopped at any time using the **SEPStop[Event]** service where **Event** could be any of the following:

a. **DREvent**
b. **Price**
c. **Message**

Every SEP causes ISY to publish the state of that event to all subscribers. Please note that the published events follow the same schema as those in the WSDL as well as **sepobjs.xsd** for requests and namely:

a. **sepobjs.xsd**.**SEPDRObject**
b. **sepobjs.xsd**.**SEPMessageObject**
c. **sepobjs.xsd**.**SEPPriceObject**

For more information on the type of events published by ISY, please see section 4.

All events are stored in the file system and therefore are reevaluated at startup and go through their normal life cycle.

2.2 Lifecycle of DR Events

Unlike straight forward nature of Price, Message, and Meter events, DR events a little different since they provide information as to how the HAN should respond. For instance, an event could target a specific Device Class such as Pool Pumps or a list of device classes. Or, Duty Cycle requires ISY to already know the total period during which Duty Cycle percent of the total time the device should be off and the rest on. As such, certain configuration options should be made to the operational environment of ISY.

Before an event can be acted upon, ISY goes through all the configured devices/nodes in the system and performs the following evaluation to filter out those nodes/devices that do not meet the criteria for the DR event:

Whether or not the node's Device Class (as defined by deviceClass in Node: see ISY WSDK Developer's Guide) is included in the event's Device Class. See section 2.2.1 for more information on Device Class.

2.2.1 Device Class

Device class defines the class of a device and is represented by the **deviceClass** element in **sepobjs.xsd**.**SEPDRObject**. Device classes are defined in **sepobjs.xsd**.**SEPDeviceClass** type as follows:

```
1           = HVAC / Thermostats
2           = Strip Heater
4           = Water Heater
8           = Pool Pump
16          = Smart Appliance
32          = Irrigation Pump
64          = Managed Load
128         = Simple
256         = Exterior Lighting
512         = Interior Lighting
1024        = Electric Vehicle
2048        = Generation System
4096        = Washer
8192        = Dryer
16384       = Oven
32768       = Refrigerator
65535       = ALL (0)
```

For multiple device classes, the values for each class are OR'ed with one another (e.g. 64 | 128).

To set the Device Class for a node, please use ISY Web Service: **SetNodePowerInfo(deviceClass, wattage, dcPeriod)**.

2.2.2 Duty Cycle

Duty Cycle is the percentage of total period of time during which the device must remain off. For instance a Duty Cycle of 50% for a total period of 1 hour means that the device is turned Off for 30 minute and then turned on for 30 minutes.

Duty Cycle is defined by the **dutyCycle** element in **sepobjs.xsd**.**SEPDRObject** and the value is in percentage (0-100).

ISY uses the following rules to figure out whether or not a device can be put on Duty Cycle:

The device in question has a non-zero dcPeriod (defined by the **dcPeriod** attribute in the Node object: see ISY Developer's Guide).

2.2.3 Opting In/Out

DR events must explicitly be acted upon by the user. In SEP vernacular, the actions taken by the user are called Opting in or Opting out.

In case, Auto Opt-in has been set, ISY automatically opts the user in as soon as the events arrive (see section 3).

SEPDROpt service enables you to opt in/out of a specific DR event. In case of opting out, the event goes into the Done state and will not be reevaluated again. In case of opting in:

a. If the event is in the **Scheduled** state, it remains in the same state but an opt in event is sent out to ISY subscribers and the utility through the meter
b. If the event is in the **Running** state, it starts processing the event by adjusting HAN devices to match the attributes as defined by the event

2.2.4 Invalidating a DR Event

ISY automatically opts the user out of a DR event in case any changes to HAN devices invalidate the DR event's attributes.

For instance if the event calls for a Duty Cycle of 50% applied to pool pumps, and if the pool pump was **on** prior to the event, and if the pool pump should now be in the **off** cycle, and if someone turns **on** the pool pump, then ISY invalidates the DR event and automatically opts the user out.

2.2.5 Revert to the State Prior to DR Event

Using the configuration options (see section 3), you can configure ISY to go back to its state before the start of the DR Event. Please note that in the following two conditions, ISY does not honor the Revert request:

1. If the customer chooses to opt out during a running DR event
2. If the DR event is invalidated (see section 2.2.4)

3. Configuration

User configurations are simple means of letting ISY know what actions should be taken based on received SEP events.

Configurations are uploaded to ISY by posting to a specific URL in the following format:
/file/upload/[config_file_URI]?load=y

3.1 Base Electricity Configuration

Post **UserElectricityOptions** object to:
/file/upload/CONF/ELEC.CFG?load=y

3.2 Message Configuration

Post **SEPMessageUserOptions** object to:
/file/upload/CONF/EMMSO.CFG?load=y

3.3 Price Configurations

Post **SEPPriceUserOptions** object to:
/file/upload/CONF/EMPO.CFG?load=y

3.4 DRLC Configurations

Post **SEPDRUserOptions** object to:
/file/upload/CONF/EMDO.CFG?load=y

3.5 Meter Configurations

Post **SEPMeterUserOptions** object to:
/file/upload/CONF/EMMO.CFG?load=y

4. ISY SEP Events (control = "_12")

In addition to all the events published by ISY framework, ISY publishes SEP events using control = _17.

4.1 Network Status Changed (action = "1")

Status of Zigbee SEP Network.
node = null
eventInfo = **zigbee.xsd.ZigbeeNetwork**

4.2 Time Status Changed (action = "2")

Time from the Zigbee SEP meter changed.
node = null
eventInfo = **zigbee.ZigbeeSEPTime**

4.3 New Message (action = "3")

A new Message event was received and validated.
node = null
eventInfo = **sepobjs.xsd**.**SEPMessageObject**

4.4 Scheduled Message (action = "31")

A new Message event was received, validated, and scheduled for future execution.
node = null
eventInfo = **sepobjs.xsd**.**SEPMessageObject**

4.5 Message Stopped (action = "4")

Message event was stopped and is no longer active
node = null
eventInfo = **sepobjs.xsd**.**SEPMessageObject**

4.6 New Price (action = "5")

A new Price event was received and validated
node = null
eventInfo = **sepobjs.xsd**.**SEPPriceObject**

4.7 Scheduled Price (action = "51")

A new Price event was received, validated, and scheduled for future execution
node = null
eventInfo = **sepobjs.xsd**.**SEPPriceObject**

4.8 Price Stopped (action = "6")

Price event was stopped and is no longer active
node = null
eventInfo = **sepobjs.xsd**.**SEPPriceObject**

4.9 New DR Event (action = "7")

A new DR event was received and validated
node = null
eventInfo = **sepobjs.xsd**.**SEPDRObject**

4.10 Scheduled DR Event (action = "71")

A new DR event was received, validated, and scheduled for future execution
node = null
eventInfo = **sepobjs.xsd**.**SEPDRObject**

4.11 DR Event Stopped (action = "8")

DR event was stopped and is no longer active
node = null
eventInfo = **sepobjs.xsd**.**SEPDRObject**

4.12 Metering Event (action = "9")

Meter attribute value changed
node = null
eventInfo = **sepobjs.SEPMeteringReport**
Root element = none

4.13 Metering Format Event (action = "10")

Meter format attribute values changed
node = null
eventInfo = **sepobjs.SEPMeterFormat**
Root element = **MeterFormat**

5. REST Interface

REST is an easy to use URL based command set which allows the developer to communicate and control ISY's Energy Management services.

Except for uploading configuration files, all REST commands use HTTP GET method.

5.1 Message

/rest/emeter/message
Returns a list of all messages (10)
Object: **sepobjs.SEPMessageObject**

/rest/emeter/message/reset
Clears all the cached and running events

/rest/emeter/message/query
Queries the meter for the last message

/rest/emeter/message/<event_id>
Returns the message with the given event_id
Object: **sepobjs.SEPMessageObject**

/rest/emeter/message/<event_id>/confirm
Sends a confirmation to for the message with the given event_id

/rest/emeter/message/log
Returns the log for messages (see section 6.0 for format of the log)

5.2 Price

/rest/emeter/price
Returns a list of all prices (10)
Object: **sepobjs.SEPPriceObject**

/rest/emeter/price/reset
Clears all the cached and running events

/rest/emeter/price/query
Queries the meter for the current price if any

/rest/emeter/price/query?offset=num1&*maxListing*=num2
Queries the meter for the scheduled prices if any:
offset – a number between -255 to 255 representing the number of minutes from the current time
maxListing – a number from 0 to 9 (0 returns means all)

/rest/emeter/price/<event_id>
Returns the price with the given event_id
Object: **sepobjs.SEPPriceObject**

/rest/emeter/price/log
Returns the log for prices (see section 6.0 for format of the log)

5.3 DRLC

/rest/emeter/drlc
Returns a list of all DRLC events (10)
Object: **sepobjs.sepo.SEPDRObject**

/rest/emeter/drlc/reset
Clears all the cached and running events

/rest/emeter/drlc/query?*offset*=num1&*maxListing*=num2
Queries the meter for the scheduled DRLC events if any:
offset – A number between -65535 to 65535 representing the number of minutes from the current time
maxListing – a number from 0 to 9 (0 returns means all)

/rest/emeter/drlc/<event_id>
Returns the DRLC event with the given event_id
Object: **sepobjs**.**SEPDRObject**

/rest/emeter/drlc/<event_id>/opt_in

Opts in to the DRLC event with the given event_id

/rest/emeter/drlc/<event_id>/opt_out
Opts out of the DRLC event with the given event_id

/rest/emeter/drlc/log
Returns the log for DRLC events (see section 6.0 for format of the log)

5.4 Zigbee SEP (Metering)

/rest/emeter
Summary of Zigbee network, module, DRLC, Price, Message, and default Metering attributes:
Object: **sepobjs.SEPSummary**
Root element: **AMIMeterSummary**

/rest/emeter/nework
Returns the current status of Zigbee Network.
Object: **zigbee.xsd.ZigbeeNetwork**
Root element: **ZBNetwork**

/rest/emeter/module
Returns the definition of currently installed Zigbee Module.
Object: **zigbee.xsd.ZigbeeModule**
Root element: **Module**

/rest/emeter/module/reset
Factory resets (to default) the currently installed Zigbee Module

/rest/emeter/time
Returns the module's current time and whether or not the module has been synchronized with the ESI/Meter.
Object: **sepobjs.xsd.ZigbeeSEPTime**
Root element: **Time**

/rest/emeter/time/query
Queries the ESI/Meter for current time and synchronizes the module

/rest/emeter/format
Returns the formatting information for measurements
Object: **sepobjs.xsd:SEPMeterFormat**
Root element: **MeterFormat**

/rest/emeter/format/query

Queries the meter formatting information

/rest/emeter/metering

Returns all the retrieved operational and reporting attributes from the meter
Object: **sepobjs.xsd:SEPMeteringReport**
Root element: **AMIMetering**

/rest/emeter/metering/log

Returns the log for meter events (see section 6.0 for format of the log)

/rest/emeter/query

Queries the meter for default attributes. The defaults may be different based on requirements from each utility provider

/rest/emeter/query/attr

Queries up to 5 specific attributes of the meter :
/rest/emeter/query/attr/***attribute_id_1***[/attribute_id_2[_5]]
Where ***attribute_id_x*** is defined in **sepobjs.SEPMeteringAttribute**

6. Logs

Each event type has its own log which can be retrieved by using the respective REST interface.

All logs are tab delimited and end with a new line (\n) and follow the following format for the first 4 elements:

a. **Time Format:** M = Military Time, A = AM/PM
b. **Time of Event:** when the event was received or its state changed
c. **Type of Entry:** Log, Error, System Startup
   In case of Error/System Startup, the next entry is text
   In case of Log, the next entries depend on the type of event defined below
d. **Source of Event:** B = Broadband, M = Meter, A = Any

6.1 Message

/rest/emeter/message/log

/rest/emeter/message/log?reset=true

Clears the message log

In case of Error and System Startup, this entry is Text.

In case of Log, the following additional attributes are tab delimited:

a. **Event ID**
b. **Event State:** [Expired | Done | Scheduled | Running | Restored]
c. **Status:** [Confirmed | Unconfirmed | NA]
d. **Start Time**
e. **Duration:** in minutes
f. **Priority:** [Low | Medium | High | Critical | NA]
g. **Requires Confirmation:** [true | false]
h. **Message:** text

6.2 Price

/rest/emeter/price/log

/rest/emeter/price/log?reset=true

Clears the price log

In case of Error and System Startup, this entry is Text.

In case of Log, the following additional attributes are tab delimited:

a. **Event ID**
b. **Event State:** [Expired | Done | Scheduled | Running | Restored]
   ----- *If Expired/Done, the entry ends here. Otherwise:*
c. **Start Time**
d. **Duration:** in minutes
e. **Currency:** [USD | CAD | NA]
f. **Unit of Measure:** kWh
g. **Trailing Digits**
h. **Price**
i. **Number of Tiers**
j. **Tier**
k. **Label**

6.3 DRLC

/rest/emeter/drlc/log

/rest/emeter/drlc/log?reset=true

Clears the DRLC log

In case of Error and System Startup, this entry is Text.

In case of Log, the following additional attributes are tab delimited:

a. **Event ID**
b. **Event State:** [Expired | Done | Scheduled | Running | Restored]
c. **Opt Status: [**Unconfirmed | Opted In | Opted Out | NA]
   ----- *If Expired/Done, then only Stop Reason attribute follows and the entry ends. Otherwise Stop Reason is removed and the entry continues with the Start Time*
d. **Stop Reason: [**Completed | Canceled | Superseded | Opted Out | NA]
e. **Start Time**

f. **Duration:** in minutes
g. **Enrollment Group**
h. **Device Class:** All if all device classes otherwise the number
i. **Criticality:** [Unknown | Green | 1 | 2 | 3 | 4 | 5 | Emergency | Planned Outage | Service Disconnect | NA]
j. **Cooling Offset**
k. **Heating Offset**
l. **Cooling Setpoint**
m. **Heating Setpoint**
n. **Average Load Adjustment**
o. **Duty Cycle**

6.4 Meter

/rest/emeter/metering/log

/rest/emeter/metering/log?reset=true

Clears the Metering log

In case of Error and System Startup, this entry is Text.

In case of Log, the following additional attributes are tab delimited:

a. Attribute Name
For a list of attribute values, please peruse ***sepobjs.SEPMeteringAttribute***

b. Attribute Value
Numeric value