



ISY
OpenADR (VEN)
Developer's Manual

Web Services SDK and REST Interface
Based on firmware 4.2.30

TABLE OF CONTENTS

0.0 REVISION HISTORY	3
1. INTRODUCTION	4
2. PUTTING IT TOGETHER	5
2.1 Retrieve Configuration	5
2.2 Save Configuration	5
2.3 Retrieve Reporting Configuration (2.0b)	6
2.4 Save Reporting Configuration (2.0b)	6
2.5 Save Opt Schedules (2.0b)	6
2.6 Get notified of OpenADR Events	7
action = "1" → OpenADR Error	7
action = "2" → OpenADR Status Update	7
action = "8" → OpenADR Registration Status	7
action = "9" → OpenADR Report Status	7
action = "10" → OpenADR Opt Status	7
2.7 Retrieving OpenADR 2.0 Full Payload	8
2.8 Opting In and Out of Events (2.0a/2.0b)	8
2.9 Clearing All Events	8
2.10 Registration Service URLs (2.0b)	8
2.11 Report Service URLs (2.0b)	9
2.12 Opt Service URLs (2.0b)	9
2.13 Push URLs	10

0.0 Revision History

Date/Firmware	Type	Change	Description
2014/9/22 4.2.15	DOC	Change	Fixed heading for section 2.11
2014/1/13 4.1.2	WSDK	New	Opt Services Added optId to eiEvent
2013/11/08 4.1.1	WSDK	Change	In OADRReportsStatus: Changed lastReportedTime to nextReportTime Changed lastSampledTime to nextSampleTime Changed registration URLs and added /party/reregister
2013/11/04 4.1.0	WSDL	NEW	2.0b Reporting support
2013/08/12	WSDL	Change	Support for 2.0b - Oadrobjs.xsd: added RegistrationID UseXmlSignatures - Registration ops URLs
2013/06/01 4.0.6	INITIAL	INITIAL	INITIAL

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) and OpenADR 2.0(a) and (b).

OpenADR is a set of standardized messages which convey load, price, and other desired events from the utility to the facility. In OpenADR vernacular, the server sending the events is called a VTN and the recipient of the events is called a VEN. ISY is a VEN.

ISY can natively process and orchestrate the OpenADR events/instructions. To help expedite OpenADR application development based on ISY, complex OpenADR messages/events are transformed to simple XML payloads that can be understood easily. The schema for the transformation is captured in [oadrobjects.xsd](#).

2. Putting it Together

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 OpenADR (21010) 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.

For more information on finding ISY on the network and communications thereto, please consult ISY WS-SDK Manual.

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

2.1 Retrieve Configuration

Configuration is represented by `oadr:AutoDRConfig` element (defined in `oadrobjects.xsd`). To retrieve configuration, you need to send a `GetSysConf` SOAP request to retrieve `/CONF/ADR.CFG`, an example of which is as follows:

```
POST /services HTTP/1.1\r\n
Host: 192.168.0.112:80\r\n
Authorization: Basic YWRtaW46YWRtaW4=\r\n
Content-Length: 157\r\n
Content-Type: text/xml; charset="utf-8"\r\n
SOAPACTION:"urn:udi-com:service:X_Insteon_Lighting_Service:1#GetSysConf"
\r\n\r\n
<s:Envelope><s:Body><u:GetSysConf xmlns:u="urn:udi-
com:service:X_Insteon_Lighting_Service:1"><name>/CONF/ADR.CFG</name></u:GetSysConf
></s:Body></s:Envelope>
```

2.2 Save Configuration

To save OpenADR Configuration, an instance of `oadr:AutoDRConfig` element (defined in `oadrobjects.xsd`) must be saved into `/CONF/ADR.CFG` an example of which is as follows:

```
POST /file/upload/CONF/ADR.CFG?load=y HTTP/1.1\r\n
Host: 192.168.0.112:80\r\n
Authorization: Basic YWRtaW46YWRtaW4=\r\n
Content-Length: 400\r\n
Content-Type: text/xml; charset="utf-8"\r\n
\r\n\r\n
<AutoDRConfig>
....
</AutoDRConfig>
```

2.3 Retrieve Reporting Configuration (2.0b)

Configuration is represented by oadr:OADRReportsConfig element (defined in oadrobjs.xsd). To retrieve configuration, you need to send a GetSysConf SOAP request to retrieve **/CONF/ADRR.CFG**, an example of which is as follows:

```
POST /services HTTP/1.1\r\n
Host: 192.168.0.112:80\r\n
Authorization: Basic YWRtaW46YWRtaW4=\r\n
Content-Length: 157\r\n
Content-Type: text/xml; charset="utf-8"\r\n
SOAPACTION:"urn:udi-com:service:X_Insteon_Lighting_Service:1#GetSysConf"
\r\n\r\n
<s:Envelope><s:Body><u:GetSysConf xmlns:u="urn:udi-
com:service:X_Insteon_Lighting_Service:1"><name>/CONF/ADRR.CFG</name></u:GetSysCon
f></s:Body></s:Envelope>
```

2.4 Save Reporting Configuration (2.0b)

To save OpenADR Reporting Configuration, an instance of oadr:OADRReportsConfig element (defined in oadrobjs.xsd) must be saved into **/CONF/ADRR.CFG** an example of which is as follows:

```
POST /file/upload/CONF/ADRR.CFG?load=y HTTP/1.1\r\n
Host: 192.168.0.112:80\r\n
Authorization: Basic YWRtaW46YWRtaW4=\r\n
Content-Length: 400\r\n
Content-Type: text/xml; charset="utf-8"\r\n
\r\n\r\n
<AutoDRConfig>
...
</AutoDRConfig>
```

2.5 Save Opt Schedules (2.0b)

To save OpenADR Opt Schedules, an instance of oadr:OADROptSchedules element (defined in oadrobjs.xsd) must be saved into **/CONF/ADROPT.CFG** an example of which is as follows:

```
POST /file/upload/CONF/ADRR.CFG?load=y HTTP/1.1\r\n
Host: 192.168.0.112:80\r\n
Authorization: Basic YWRtaW46YWRtaW4=\r\n
Content-Length: 400\r\n
Content-Type: text/xml; charset="utf-8"\r\n
\r\n\r\n
<OADROptSchedules>
...
</OADROptSchedules>
```

2.6 *Get notified of OpenADR Events*

ISY is based on publish subscribe framework and, therefore, the client can subscribe to ISY and then be notified of all events from then onwards and without the need to keep polling ISY.

To learn about subscribing to ISY, please consult ISY WS-SDK Manual section 4.3.

When subscribed, events with *Control*=_10 and the following actions are those that relate to OpenADR:

action = "1" → OpenADR Error

node = null
eventInfo = null

action = "2" → OpenADR Status Update

node = null
Namespace: **oadr** (oadrobjs.xsd)
<eventInfo>
 oadr:OpenADRState
</eventInfo>

action = "8" → OpenADR Registration Status

node = null
Namespace: **oadr** (oadrobjs.xsd)
<eventInfo>
 oadr:OADRRegistration
</eventInfo>
*Profile 2b only

action = "9" → OpenADR Report Status

node = null
Namespace: **oadr** (oadrobjs.xsd)
<eventInfo>
 oadr:OADRReport
</eventInfo>
*Profile 2b only

action = "10" → OpenADR Opt Status

node = null
Namespace: **oadr** (oadrobjs.xsd)
<eventInfo>
 oadr:oadrOpt
</eventInfo>
*Profile 2b only

2.7 Retrieving OpenADR 2.0 Full Payload

OpenADRState which is published as event to clients, shows only the overall status of OpenADR subsystem (or any changes thereto).

For Profile 2(a), it's usually very important to know more details especially about Signals and Intervals. Since the payload for OpenADR 2.0 profile is much larger than the amount of information ISY wishes to publish to clients, it's best for clients to retrieve the payload if and only if the information in the event is not sufficient.

To do so, an http GET on:
/rest/oadr

To help expedite OpenADR application development based on ISY, complex OpenADR messages/events are transformed to simple XML payloads that can be understood easily. As such, the result of /rest/oadr is an oadr:eiEvents element the schema for which is captured in [oadrobjects.xsd](#).

2.8 Opting In and Out of Events (2.0a/2.0b)

You can opt in and out of events.

To Opt-In, do a GET on:
/rest/oadr/<event-id>/optin

To Opt-Out, do a GET on:
/rest/oadr/<event-id>/optout

2.9 Clearing All Events

To clear all events, do an http GET on:
/rest/oadr/clear

2.10 Registration Service URLs (2.0b)

Register to VTN
/rest/oadr/party/register

Cancel Registration to VTN
/rest/oadr/party/unregister

Query Registration
/rest/oadr/party/queryreg

Re-Registration
/rest/oadr/party/reregister?type=val
Where type is *refresh* | *renew*
refresh = uses the existing Registration ID
renew = does not use the existing Registration ID

2.11 Report Service URLs (2.0b)

Register report Meta Data
/rest/oadr/report/register?type=val
Where type is *usage* | *status* | *history* | *all* [default]

Get status of all reports (oadrobjects:OADRReportsStatus)
/rest/oadr/report/status

2.12 Opt Service URLs (2.0b)

Retrieve all schedules (oadrobjects:OADROptSchedules)
/rest/oadr/opts

Cancel an Opt Schedule
/rest/oadr/opts/cancel/<opt-id>
Where opt-id is the unique id for the opt to be canceled (oadrobjects:oadrOpt)

Cancel all Opt schedules
/rest/oadr/opts/cancelAll

Clear all inactive Opt schedules
/rest/oadr/opts/clear

2.13 Push URLs

For 2.0a Push mode, ISY endpoint is:

/OpenADR2/Simple/EiEvent

For 2.0b Push mode, ISY endpoint is:

/OpenADR2/Simple/2.0b/[Service]

Where service is:

EiEvent – Event services

EiReport – Reporting services

EiOpt – Scheduling services