ISY
Portal Integration
Developer's Manual

Based on firmware 4.0.5

TABLE OF CONTENTS

0.0 Revision History

Date/Firmware	Type	Change	Description
2012/03/29 **3.2.2**	DOC	UPDATE	Using portal defined <URLBase>
2012/03/16 **3.2.0**	DOC	NONE	
2011/12/13 **3.1.15**	DOC	UPDATE	Clarification for **Authenticate and Subscription**
2011/11/29 – **3.1.14**		NEW	Initial Revision

1. Introduction

ISY is an automation and energy management platform with well-defined Web Services and REST interfaces that provide the developers with a complete set of management and event handling functionality. For information on Web Services/REST/Events please consult ISY WSDK Development Guide.

Since ISY has its own web server thus network traffic must be forwarded to the ports (HTTP and HTTPS) that the web server listens to. For local access, this process is as trivial as pointing the client to ISY's IP address/web server port combination. On the other hand, for remote access a port forwarding rule must be created in the router to forward Internet traffic to ISY's IP address/web server port which might become a daunting task for anyone not intimately knowledgeable about routers and networking.

ISY's Portal Integration Module enables integration of Cloud based/Portal services to act as proxy for communications with ISY. With Portal Integration Module, ISY makes a persistent outbound connection to a predefined server the socket for which can be used to proxy client requests back to ISY and using the same Web Services and REST interfaces. In this manner, then, ISY will be accessible remotely without the need for port forwarding rules in the router.

Please note that Portal Integration Module is project/portal specific and has to be approved by UDI before going to production. If you are interested in developing a Portal solution for ISY, please contact sales@universal-devices.com and we'll activate a development Portal Integration Module for your authorized ISYs.

2. Theory of Operation

Portal Integration Module is designed to be as simple, efficient, and transparent as possible. The module depends on the following 3 different URLs:

2.1 Dispatcher URL

Dispatcher URL is the URL for a server which may be able to provide some level of load balancing and is the first URL/server to which ISY tries to connect to. The URL must support HTTPS GET method and must return (perhaps based on some load balancing algorithm) a list of one or more Proxy Servers (see section 2.2) separated by a newline (\n) such as:

https://www1.universal-devices.com/isy
https://www2.universal-devices.com/isy
….
and so no and so forth.

Dispatcher URLs are different for each specific Portal Integration Module and, upon approval, are built into ISY's firmware.

This said, for development purposes, Dispatcher URL can be modified through the shell by the following commands

SPU – Sets the Dispatcher URL to a user defined value

DPU – Reverts the Dispatcher URL to the Default value as defined by that specific Portal Integration Module

2.2 Proxy URL

Proxy URL is the actual URL to which ISY makes a connection to, keeps the socket open indefinitely (or till there's a connection error), and is defined as one of the entries in the output of doing a GET on the Dispatcher URL (see section 2.1).

It's up to the server to make and maintain a mapping between the socket connection and a specific ISY. In most cases, it's best to use ISY's MAC/UUID address as the unique identifier for the mapping since the MAC address uniquely identifies an ISY and never changes. Please see section 2.4 for information on how to retrieve ISY's MAC address.

***Important Note*:** If the portal is going to proxy Admin Console requests, please do make sure that there's an **/isy** *anywhere* in the proxy URL.

2.3 Subscription URL

Subscription URL is the URL to which ISY publishes all its events. Subscription URL should be provided in the ***<reportURL>*** element of ***Subscribe*** Web Service call by the Proxy Server. Subscription URL can be an entirely different URL than the Proxy URL and it must support HTTPS.

There can only be one subscription to the portal the SID (Subscription ID) for which is always ***0***.

Please review ISY's WSDK for more information on subscriptions and specifically section 6.2 for **/rest/subscriptions** interface information.

2.4 Authentication and Authorization

For ISY, the Portal is a preauthorized and authenticated client and thus ISY does not check credentials for requests coming from the portal. As such, it's of utmost importance to make sure the Proxy Server has a robust authentication, authorization, and security infrastructure.

Since the proxy is acting on behalf of a client, as such the proxy **must** trap the ***Authenticate*** Web Service (from clients) and process its own authentication and authorization.

Furthermore, the proxy must be aware of URLs which do not require any authentication:

1. ***/desc***
2. ***/WEB/****

Note: If you are going to let the Admin Console communicate with ISY through the proxy and ***if and only if*** you wish the Admin Console to use a portal-defined ***URLbase*** to communicate with the proxy, then you must trap the ***/desc*** URL, and replace the ***<URLBase>*** element with the URL bases of your own choosing. Example:

Replace the following:
<URLBase>http://192.168.0.129</URLBase>

With:
<URLBase>https://your-portal-url/[any-path]+</URLBase>

2.5 Subscription

Since the clients are now going through the proxy server, as such the proxy server must now provide subscription services to the clients. Therefore, the proxy server must trap subscription requests and stop them from getting to ISY. The following Web Services must be trapped:

1. ***Subscribe***
2. ***IsSubscribed***

For definition of these services, please consult the WSDK documentation.

2.6 Putting It Together

The process flow is as follows:

1. Upon reboot
    a. ISY tries to retrieve a list of Proxy Servers from the Dispatcher URL
    b. For each Proxy URL ISY tries to make a connection thereto
    c. If successful, a session is established; **go to 2**
    d. If unsuccessful, try the next URL
    e. If all URLs fail, wait for 1 minute and **go back to a**

2. The Proxy Server accepts the connection from ISY and starts communicating with ISY using ***this socket*** here on out:
    a. Proxy Server retrieves ISY identification information by submitting an HTTP GET request to **/rest/whoami** (see Section 3)
    b. Proxy Server uses the MAC address in addition or combination with some other identifying information to create and maintain a mapping between the socket and ISY
    c. Proxy Server then subscribes to ISY while providing it a unique URL for the ***<reportURL>*** element of the ***Subscribe*** Web Service call.

    Important Note: You cannot use REUSE_SOCKET value for <reportURL> since it would cause ISY to use the same socket for both subscriptions as well as command/control

3. Proxy Server uses some algorithms based on ISY's Heartbeat Event in combination with other control information (such as TCP KeepAlive packets sent by ISY or periodic ***/rest/whoami*** requests) to make a determination as to whether or not ISY is still communicating with it:
    a. In case it has determined that ISY is no longer publishing events, Proxy Server uses ***IsSubscribed*** Web Service call using ***SID of 0***, to see whether or not it's still subscribed to ISY
    b. If it's not subscribed, ***repeats 2a – 2c***
    c. If it cannot communicate with ISY at all then it closes the socket and removes all the mappings between the socket an ISY. Proxy Server shall then wait indefinitely for ISY to try and contact it again and, once contacted, ***repeats 2a – 2c***

4. Just as any other client, ISY publishes all events to the Subscription URL
    a. In case of connection/network failure, ISY will retry **3** times. In case all 3 attempts fail, ISY expires the subscription
    b. It's up to the Server to notice lack of events coming from ISY and re-subscribe (***3a)***
    c. The relationship between Subscription socket and Proxy socket is as follows:

i. If ISY can no longer communicate with the Proxy, ISY expires the associated subscription (if any)
ii. If ISY can no longer communicate with the Subscription Server, ISY only expires the subscription but the Proxy socket remains active (**4a**)

Once a socket session is established between ISY and the Proxy Server, all Web Services and REST interface calls can be issued ***as-is*** by the Proxy Server and through the Proxy socket.

The only limitation is that the commands must be issued synchronously: only one Web Service or REST interface call can be issued at any given time. i.e the Proxy Server has to wait for one command to finish before issuing the next. For best performance, it's advised that the Proxy Server uses a queuing system to queue commands and process them one at a time.

3. REST Interface

As mentioned before, the Proxy Server can use all ISY Web Services and REST interface calls, as-is (and without the Authorization headers) through the Proxy Server socket. In addition to ISY services, the Portal Integration Module provides the following additional REST interface(s):

3.1 /rest/whoami

This interface returns uniquely identifying attributes of the given ISY to which the Proxy Server is connected:

<iam>
<partner>*A string representing the partner ID*</**partner**>
<type>*A string representing the type of ISY*</**type**>
<product>*ISY's product ID which represents a model number*</**product**>
<id>*Hexadecimal MAC address*</**id**>
<SID>
Subscription ID if any … this element will not be there if ISY has no active subscription to the Portal. Otherwise, the value is always: uuid:0
</**SID**>
<electricity>
<providerId>
A String representing the Utility provider ID if any
</**providerId**>
<enrollmentGroup>
An Integer representing the Utility enrollment group if any
</**enrollmentGroup**>
</**electricity**>
</**iam**>

Example:

```
<iam>
        <partner>Portal-X-Partner</partner>
        <type>ISYv30</type>
        <product>1110</product>
        <id>0021b9030405</id>
        <SID>uuid:0</SID>
        <electricity>
                <providerId>SDGE</providerId>
                <enrollmentGroup>0</enrollmentGroup>
        </electricity>
</iam>
```