ISY
Node Server
Developer's Manual

REST Interface
Based on firmware 4.5.1 Alpha

TABLE OF CONTENTS

Revision History

Date	Firmware	Description
2015/05/12	4.5.1	Clarifications of APIs and NLS files
2015/04/12	4.5	Initial

1. Introduction

ISY is an award winning platform for automation and energy management. With the introduction of Node Servers, the ISY now supports any protocol implemented by a third party in much that same way that INSTEON, Z-Wave and Zigbee are supported.

The concepts remain the same. The big difference is that instead of the ISY generating the events and running device commands, the node server does.

2. What is a Node?

A node represents all, or a subset of, a physical device such as lamp, switch and keypad, smoke detector, etc., or a conceptual device such as weather information or even stock quotes.

A *node definition* is used to describe a node. It contains the list of status values it maintains (e.g. the current temperature, heat/cool setpoints for a thermostat), the list of commands it accepts (e.g. on/off for a dimmer lamp), and a list of the commands it may send out (e.g. on/off for a dimmer switch).

A node server simply defines the set of nodes its supports, and provides the REST services to support them.

3. Node Server Configuration on ISY

3.1 Files

/editor	Contains all the XML editors files (.xml)
/nodedef	Contains all the XML node definitions (.xml)
/nls	Contains all the NLS properties files (.txt)
/version.txt	Contains the version of these files

These files are normally supplied as a .zip file by the node server developer and installed by the user through the ISY Admin console. In each directory, one or more files may be used. All filenames are restricted to 8.3 format.

If the node server developer creates a new version of the files, they can be installed over the old ones on the ISY. It is up to the node server developer to ensure any required backwards compatibility of nodes.

e.g. Example Zip File contents
/editor/edit.xml

/nodedef/ndef.xml
/nls/EN_US.txt
/version.txt

/editor

An editor defines the parameters for a widget in the client, such as a combobox, a numeric field etc. It defines the set of values and the unit(s) of measure available. An editor may contain multiple <*range*> entries, each of which must have a unique UOM.

/nodedef

A node definition defines the status and commands available to a node.

/nls

A single NLS file is used for each language. The naming convention is <*language*>_<*countryCode*>.*txt (*e.g. *en_US.txt* for USA English)
NLS is a set of name/value pairs used to display values in natural language (such as English).

3.2 Network Connection

3.2.1 From Isy to Node Server

The REST API is used to communicate with a node server when using a network connection. The ISY uses basic authentication with either http or https to communicate with the node server. A custom base URL is also prepended to the REST command, allowing the node server to customize the location of its REST support.

For example, if a base URL of */nodeservers/joe* is configured, then the following URL would be sent to the node server to query a node:

/nodeservers/joe/nodes/<*nodeAddress*>**/query**

Having a base URL also allows a device to support multiple node servers, each with its own unique base URL.

3.2.2 From Node Server to Isy

The node server must use basic authentication with either http or https to communicate with the ISY. It must also know the *profile number* the node server has been assigned on the ISY because most REST API calls require this number in the URL. The ISY uses the profile number to ensure only the nodes owned by the profile can be modified, and to choose the ISY user number the node server should be using.

For example, if the node server has been assigned profile number 5, then something like the following URL would be used to update device status in ISY:

/rest/ns/5/nodes/n005_dimmer_2/report/status/ST/25.2/percent

3.2.3 Responses

When a Node Server receives a REST command, one of the following responses must be sent out immediately, ***before*** processing the request. The ISY will send a similar response ***after*** processing a request.

200 - HTTP_OK
Valid request received, will run it

404 - HTTP_NOT_FOUND
Unrecognized request received and ignored.

503 - HTTP_SERVICE_UNAVAILABLE
Valid request received but ignored because system too busy to run it

If the userid/password is missing or incorrect

401 - HTTP_UNAUTHORIZED
User authentication failed

3.3 Serial Connection

Support may be added at a later time for node servers using serial connections.

4. Required API support in Node Server

4.1 General

Each node server is required to support a set of APIs that the ISY will use to manage the nodes being supplied by the node server. Primarily, these APIs are used to add/delete/rename nodes, send commands to nodes, and request node information. Other APIs request the node server to install or upgrade itself on the ISY, and generally manage the configuration of the node server.

4.1.1 Request IDs

<*base*>/*...*[?requestId=<*requestId*>]

On most API calls, the ISY can optionally supply a *requestId*. If a *requestId* appears on the URL then the node server must send a success or fail message back to the ISY after it has completed the requested action, and, **after** all messages from that completed action have been sent to the ISY.

This allows the ISY to run a command synchronously. For example, the ISY may need to query a device and use the results of the query to do some additional processing.

4.1.2 Node Addresses

All node addresses are given a prefix assigned by the ISY. The prefix is unique to the node server thus guaranteeing that all node addresses on the ISY are unique.

The format of the node address prefix is:
n*aaa***_**

Where *aaa* is the profile number assigned to the node server in the ISY. A node address is made up of any combination of lowercase letters, numbers, and '_' character.

A node address for profile 5 could look something like:
n005_dimmer_3_1

The **dimmer_3_1** portion of the node address is completely defined by the node server or the user creating the node.

The maximum node length (including the prefix) is 19 characters.

4.2 Install

*<base>***/install/***<profileNumber>*

Instructs the node server to install all the profile files for the node server (rather than having the user do it through the ISY admin console). This is done by removing the old files and then adding all the files one by one, as follows:

- /rest/ns/<*profileNumber*>/profile/remove
- For each file:
    - /rest/ns/<*profileNumber*>/profile/upload/<*dir*>/<*filename*>
- /rest/ns/<*profileNumber*>/profile/reload

NOTE: In the current implementation, the ISY must be restarted for the new files to take effect.

4.3 Query node

*<base>***/nodes/***<nodeAddress>***/query[?requestId=***<requestId>***]**

The node server must query the specified node, and send the results to the ISY using the Report Status Rest command.

If a requestId is specified, the status of the request must be sent to the ISY after all other messages are sent.

If a <*nodeAddress*> of "**0**" is specified, then all nodes must be queried.

4.4 Get Node Status Values

*<base>***/nodes/***<nodeAddress>***/status[?requestId=***<requestId>***]**

The node server sends the current status values for the specified node to the ISY using the Report Status Rest command.

If a requestId is specified, the status of the request must be sent to the ISY after all other messages are sent.

If a <*nodeAddress*> of "**0**" is specified, then statis for all nodes must be sent.

4.5 Add All Nodes

<*base*>/add/nodes[?requestId=<*requestId*>]

Instructs the node server to add all of its nodes to the ISY (see Node Management).

If a requestId is specified, the status of the request must be sent to the ISY after all other messages are sent.

4.6 Reports from ISY

Reports provided by the ISY give the node server an opportunity to update its own database of nodes.

<*base*>/nodes/<*nodeAddress*>/report/add/<*nodeDefId*>?primary=<*node Address*>&name=<*nodeName*>

- Reports to the node server that the given node was added to the ISY.

<*base*>/nodes/<*nodeAddress*>/report/remove

- Reports to the node server that the given node was removed from the ISY.

<*base*>/nodes/<*nodeAddress*>/report/rename?name=<*nodeName*>

- Reports to the node server that the given node was renamed in the ISY.

<*base*>/nodes/<*nodeAddress*>/report/enable

- Reports to the node server that the given node was enabled in the ISY.

<*base*>/nodes/<*nodeAddress*>/report/disable

- Reports to the node server that the given node was disabled in the ISY.

NOTE: In the future, there may be additional APIs added that allow the node server more control over the actual creation and modification of nodes.

4.7 Run a command

<*base*>/**nodes**/<*nodeAddress*>/**cmd**/<*command*>
<*base*>/**nodes**/<*nodeAddress*>/**cmd**/<*command*>/<*value*>
<*base*>/**nodes**/<*nodeAddress*>/**cmd**/<*command*>/<*value*>/<*uom*>

[?<*p1*>.<*uom1*>**=**<*val1*>**&**<*p2*>*...*][**requestId=**<*requestId*>**]**

The node server must run the specified command for the specified node. This command may have originated from an ISY program, the standard ISY REST API, the admin console, or any other client. The commands normally sent are those listed in the <accepts> section of the node definition used for the given node.

The numeric value of the UOM is always supplied and is never one of the common names. For example, **51** will be used instead of '**percent**'. For parameters in the <*pX*>.<*uomX*> format, the numeric uom value is always prefixed by '**uom**'

If a requestId is specified, the status of the running the command must be sent to the ISY after the command has completed or failed.

nodeAddress	The full address of the node (e.g. 'n005_switch_1)
command	The command to perform (e.g. 'DON', 'CLISPH', etc.)
pN	*Nth* Parameter name (e.g. 'level')
uomN	Unit of measure of the *Nth* parameter (e.g. 'uom58')
valN	The numeric value of the *Nth* parameter (e.g. '80', '80.01' etc,)

Commands may also have an unnamed parameter

value	The value of the unnamed parameter.
uom	Unit of measure of the value of the unnamed parameter (e.g. 51)

E.g.
/myserver/nodes/n005_switch_1/report/cmd/DON
/myserver/nodes/n005_switch_1/report/cmd/DON/80/51
/myserver/nodes/n005_switch_1/report/cmd/DON?level.uom51=80
/myserver/nodes/n005_switch_1/report/cmd/DON/80/percent?rate.uom58=0.3

5. REST support in ISY

REST is an easy to use URL based command set which allows the developer to communicate with the ISY.

Unless otherwise specified, all REST commands use HTTP GET method.

If no Response is provided, then UDIDefaultResponse must be assumed:

WSDL:zw:UDIDefaultResponse

Notes:

- URL Prefix**: /rest/ns/**<*profileNumber*>**/**
- The *profileNumber* specified on the URL determines which ISY userid/password will be accepted by the ISY for the request.

5.1 Reporting status updates

/nodes/<*nodeAddress*>**/report/status/**<*driverControl*>**/**<*value*>**/**<*uom*>

Updates the ISY with the current value of a driver control (e.g. the current temperature, light level, etc.)

nodeAddress	The full address of the node (e.g. 'n005_dimmer_1')
driverControl	The name of the status value (e.g. 'ST', 'CLIHUM', etc.)
value	The numeric status value (e.g. '80.5')
uom	Unit of measure of the status value

E.g. /rest/ns/5/nodes/n005_dimmer_2/report/status/ST/25.2/percent

5.2 Reporting a command

/nodes/<*nodeAddress*>**/report/cmd/**<*command*>
/nodes/<*nodeAddress*>**/report/cmd/**<*command*>/<*value*>
/nodes/<*nodeAddress*>**/report/cmd/**<*command*>/<*value*>/<*uom*>

[?<*p1*>.<*uom1*>**=**<*val1*>**&**<*p2*>.<*uom2*>**=**<*val2*>**&**<*p3*>…**]**

Sends a command to the ISY that may be used in programs and/or scenes. A common use of this is a physical switch that somebody turns on or off. Each time the switch is used, a command should be reported to the ISY. These are used for scenes and control conditions in ISY programs.

nodeAddress	The full address of the node (e.g. 'n005_switch_1)
command	The command to perform (e.g. 'DON', 'CLISPH', etc.)
pN	*Nth* Parameter name (e.g. 'level')
uomN	Unit of measure of the *Nth* parameter (e.g. 'seconds', 'uom58')
valN	The numeric value of the *Nth* parameter (e.g. '80', '80.01' etc,)

Commands may also have an unnamed parameter

value	The value of the unnamed parameter.
uom	Unit of measure of the value of the unnamed parameter

E.g.
/rest/ns/5/nodes/n005_switch_1/report/cmd/DON
/rest/ns/5/nodes/n005_switch_1/report/cmd/DON/80/percent
/rest/ns/5/nodes/n005_switch_1/report/cmd/DON?level.percent=80
/rest/ns/5/nodes/n005_switch_1/report/cmd/DON/80/percent?rate.uom58=0.3

5.3 Node Management

/nodes/<*nodeAddress*>/add/<*nodeDefId*>?primary=<*primary*>&name=<*nodeName*>

Adds a node to the ISY. To make this node the primary, set *primary* to the same value as *nodeAddress*

nodeAddress The full address of the node (e.g. 'n005_dimmer_1')
nodeDefId The id of the node definition to use for this node
primary The primary node for the device this node belongs to
nodeName The name of the node

E.g.
/rest/ns/5/nodes/n005_dimmer_2/add/MyDimmer?primary=n005_dimmer_1&name=Dimmer 2

/nodes/<*nodeAddress*>/change/<*nodeDefId*>

Changes the node definition to use for an existing node. An example of this is may be to change a thermostat node from Fahrenheit to Celsius.

nodeAddress The full address of the node (e.g. 'n005_dimmer_1')
nodeDefId The id of the node definition to use for this node

E.g. /rest/ns/5/nodes/n005_tstat_1/change/ThermostatCelsius

/nodes/<*nodeAddress*>/remove

Removes a node from the ISY. A node cannot be removed if it is the primary node for at least one other node.

nodeAddress The full address of the node (e.g. 'n005_dimmer_1')

E.g. /rest/ns/5/nodes/n005_dimmer_2/remove

5.4 Reporting ISY Request status

/report/status/<*requestId*>/fail
/report/status/<*requestId*>/success

When the ISY sends a request to the node server, the request may contain a 'requestId' field. This indicates to the node server that when the request is completed, it **must** send a fail or success report for that request. This allows the ISY to in effect, have the node server synchronously perform tasks. This message must be sent after all other messages related to the task have been sent.

For example, if the ISY sends a request to query a node, all the results of the query must be sent to the ISY before a fail/success report is sent.

requestId The request ID the ISY supplied on a request to the node server.

E.g. /rest/ns/5/report/request/1234/success

6. Natural Language Support (NLS)

6.1 General

NLS support is defined for a node server by the set of properties files in the **/nls** subdirectory. They contain the name/value pairs used by the clients and the ISY to display commands, values, controls etc.. All NLS names must be in uppercase.

A naming convention is used to organize these values.

6.2 Naming Convention Terminology

The following table shows the various attributes from XML node definitions and editors that are used in this chapter to describe how to build the name of a particular NLS value.

<*nodedef.id*>	The 'id' attribute of a node definition. e.g. <nodeDef **id="Thermostat"** nls="tstat">
<*nodedef.nls*>	The 'nls' attribute of a node definition. e.g. <nodeDef id="Thermostat" **nls="tstat"**>
<*editor.id*>	The 'id' attribute of an editor e.g <editor **id="I_OL"**>
<*range.nls*>	The 'nls' attribute of a range e.g. <range uom="25" subset="0-32" **nls="IX_I_RR"** />
<*st.id*>	The 'id' attribute of a status e.g. <st **id="CLIHUM"** editor="I_TSTAT_HUM" />
<*cmd.id*>	The 'id' attribute of a command e.g. <cmd **id="DON"**>
<*p.id*>	The 'id' attribute of a command parameter e.g. <p **id="COLOR"** editor="I_COLOR_RGB" />

6.3 Status Names

Some status values require different names for different node types. For example, ST for a dimmer should show up as 'Lamp', but ST for a drapery motor should show up as 'Drapes'. The format and lookup order of the NLS entry is:

```
ST-<nodedef.nls>-<st.id>-NAME
ST-<st.id>-NAME
```

e.g.

```
ST-ST-NAME = Lamp
ST-MYDRAPES-ST-NAME = Drapes
```

6.4 Command Names

The format and lookup order of the NLS entry for command names is:

```
CMD-<nodedef.nls>-<cmd.id>-NAME
CMD-<cmd.id>-NAME
```

e.g.

```
CMD-DON-NAME = On
CMD-MYDRAPES-DON-NAME = Open
```

6.5 Command Parameter Names

The format and lookup order of the NLS entry for command parameter names is:

```
GEN-<p.nls>-NAME
CMDP-<nodedef.nls>-<editor.id>-<p.id>-NAME
CMDP-<editor.id>-<p.id>-NAME
```

e.g.

```
GEN-MYTIMER001-NAME = On/Off Timer
```

6.6 Other Names

node definition `ND-<nodedef.id>-NAME`

6.7 Name mapped Values (Index, Percent)

Some integer values may be displayed as names instead of numeric values. Index values (uom 25), and some percent values are commonly made into names. For example, displaying the values 0-32 for Insteon Ramp Rates is not very meaningful compared to names indicating the actual durations. 'On' and 'Off' are often displayed for percentage values, while the remaining values 1-99 are usually displayed numerically.

The format of the NLS entry for mapped values is:

```
<range.nls>-<value>
```

```
e.g.  Insteon Ramp Rates

      <range id="I_RR" uom="25" subset="1-32" nls="IX_I_RR" />

IX_I_RR-1 = 9.0 minutes
IX_I_RR-2 = 8.0 minutes
...
IX_I_RR-32 = 0.1 seconds
```

6.8 Formatting in Programs

Each line of a program is formatted and displayed in different way. Custom formatting entries are used for node conditions and commands, as follows:

6.8.1 Commands

The format and lookup order of the NLS entry for program command entries is:

```
PGM-CMD-<nodedef.nls>-<cmd.id>-FMT
PGM-CMD-<cmd.id>-FMT
```

```
e.g. (All on one line)
   PGM-CMD-DON-FMT = /level/${c}/to ${v}/ /ramprate// in ${v}/
/offtimer//, turn off ${v} later/
```

/<param.id>/param text if omitted/param text if not omitted/ [.. next parameter, …]

e.g. `/level/${c}/to ${v}/ /ramprate// in ${v}/`

/ First character defines what character to use as separator, normally '/' is used

param.id Id of the parameter (e.g. 'level')
Note: This is empty for an unnamed parameter

param text if omitted
String to show if the parameter was omitted

param text if not omitted
String to show if the parameter was specified

The string for parameter text supports the following variables:

${c}	Name of the command
${v}	Formatted value of the parameter (including UOM)
${uom}	Formatted UOM without the value
${op}	Operator used (conditions only)

6.8.1.1 Command Formatting Examples

```
Assume commands are for node 'MyDevice'

1) A command with three named parameters, 'num', 'val', 'len'

/num//${c} Parameter ${v}/ /val/default/ = ${v}/ /len// (${v} bytes)/

The following program action line would be shown for:
   ${c} = "Config", num=1, val=20, and len=4:
   [Config Parameter 1][ = 20][ (4 bytes)]
      --> "Set 'MyDevice' Config Parameter 1 = 20 (4 bytes)"

   ${c} = "Config", num=5, val=25, and len omitted:
   [Config Parameter 5][ = 25][]
      --> "Set 'MyDevice' Config Parameter 5 = 25"

   ${c} = "Device", num=5, val omitted, and len=2:
   [Device Parameter 5][default][ (2 bytes)]
      --> "Set 'MyDevice' Device Parameter 5 = default (2 bytes)"

2) A command with one unnamed parameter
   //default/${v}/

   value of unnamed param omitted
   [default] --> "Set 'MyDevice' default"

   value of unnamed param = 50 percent
   [50%] --> "Set 'MyDevice' 50%"

3) A command with no parameters shows just the command name and does
not require and PGM-xxxxx entry
   DFON   -->   "Set 'MyDevice' Fast On"

Another example:
   PGM-CMD-DON-FMT = /level/${c}/to ${v}/ /ramprate// in ${v}/
/offtimer//, turn off ${v} later/

   level=50%, ramprate=3 seconds, offtimer=5 minutes
   "[to 50%][ in 3 seconds][, turn off 5 minutes later]"
   -->  "Set 'MyDevice' to 50% in 3 seconds, turn off 5 minutes later"
```

6.8.2 Status Conditions

The format and lookup order of the NLS entry for status condition format entries is:

```
PGM-ST-<nodedef.nls>-<st.id>-FMT
PGM-ST-<st.id>-FMT
```

The value is a single ***param text*** string similar to that specified for a command parameter.

e.g.

```
PGM-ST-CLISPH-FMT = ${c} ${op} ${v}
```

If not specified, then the following is used: `${c} ${op} ${v}`

6.8.3 Control Conditions

There are no custom entries for control conditions, because currently, control conditions do not include any of the command parameters.

7. Appendix

7.1 Editors

```
<editors>
   <editor id="I_OL">
      <range uom="51" subset="0-100" />
      <range uom="56" subset="0-255" />
   </editor>
   <editor id="TEMP_C">
      <range uom="4" min="4.5" max="32" step="0.5" prec="1" />
   </editor>
   <editor id="I_RR">
      <range uom="25" subset="0-32" nls="IX_I_RR_" />
   </editor>
</editors>
```

editor	id	The name of the editor.
range	uom	The unit of measure of the value (See Units of Measure) *Note: Must be unique for each range entry in an editor*
	nls	Used for *percent (51)* and *index (25)* unit of measures only. The NLS prefix to use for the name of value. e.g. for nls="IX_I_RR" value 5, the NLS entry would be: IX_I_RR-5 = 8 seconds
range *(1)*	subset	The subset of values supported defined as a set of ranges and individual values. They must be in increasing value with no duplicates or overlap. The values are limited to positive integers. Ranges are separated by a '-', individual digits are separated by a ','. e.g. subset="0-5,7,9,11-14" means these numbers: 0,1,2,3,4,5,7,9,11,12,13,14
range *(2)*	min	The minimum value (inclusive)
	max	The maximum value (inclusive)
	step	The number to increment with each step (e.g. in a spinner type widget)
	prec	The number of decimal places to keep for the value

7.2 Node Definitions

```
<nodeDefs>
   <nodeDef id="Thermostat" nodeType="143" nls="143">
      <editors> … </editors>
      <sts>
         <st id="ST" editor="I_TEMP_DEG" />
         <st id="CLISPH" editor="I_CLISPH_DEG" />
         <st id="CLISPC" editor="I_CLISPC_DEG" />
         <st id="CLIMD" editor="I_TSTAT_MODE" />
         <st id="CLIHCS" editor="I_TSTAT_HCS" />
      </sts>
      <cmds>
         <sends>
            <cmd id="DON" />
            <cmd id="DOF" />
         </sends>
         <accepts>
            <cmd id="CLISPH">
               <p id="" editor="I_CLISPH_DEG" init="CLISPH" />
            </cmd>
            <cmd id="CLISPC">
               <p id="" editor="I_CLISPC_DEG" init="CLISPC" />
            </cmd>
            <cmd id="CLIMD">
               <p id="" editor="I_TSTAT_MODE" init="CLIMD" />
            </cmd>
            <cmd id="QUERY" />
         </accepts>
      </cmds>
   </nodeDef>
</nodeDefs>
```

<nodeDef>	id	Name of this node definition (e.g. "DimmerSwitch")
	nodetype	Optional: Isy node type (See Node Types). This must be specified for device to participate in certain modules (e.g. Open ADR)
	nls	NLS key string used to override names of commands, status and other elements.
<editors>		Editors scoped to this node
<st>	id	One of the predefined driver controls e.g. "CLISPH"
	editor	The id of the editor to use
<sends>		The commands this node can send out. Used for control conditions in ISY programs and scene controllers.
<accepts>		The commands this node accepts. Used for buttons etc. in clients, and actions in ISY programs.
<cmd>	id	Name of a command.
<p>	id	Name of a command parameter. A command may have one unnamed parameter, all others must be named.
	editor	The id of the editor to use for this parameter

	init	(Optional) id of the status value this parameter should be initialized and/or synchronized with. For example, CLISPH is both a status and a command.