ISY
ELK Integration
Developer's Manual

Web Services SDK and REST Interface
Based on firmware 4.0.5

TABLE OF CONTENTS

REVISION HISTORY 4

1. INTRODUCTION 5

2. GETTING STARTED 6

3. ELK EVENTS (CONTROL = "_19") 8

3.1 Topology Changed (action = "1") 8

3.2 Area Event (action = "2") 8

3.3 Zone Event (action = "3") 9

3.4 Keypad Event (action = "4") 9

3.5 Output Event (action = "5") 10

3.6 System Event (action = "6") 10

3.7 Thermostat Event (action = "7") 11

4. REST INTERFACE 12

URL Prefix: /rest/elk/ 12

4.1 Area Commands 12

areas/query 12

area/<*areaId*>/get/status 12

area/<*areaId*>/cmd/bypass?code=<*accessCode*> 12

area/<*areaId*>/cmd/unbypass?code=<*accessCode*> 12

area/<*areaId*>/cmd/display?text=<*eText*>&beep=<boolean> &offTimerSeconds=<*seconds*> 13

area/<*areaId*>/cmd/display?id=<*notifyId*>&beep=<boolean> &offTimerSeconds=<*seconds*> 13

area/<*areaId*>/cmd/arm?armType=<*elkArmType*> &code=<*accessCode*> 13

area/<*areaId*>/cmd/disarm?code=<*accessCode*> 13

4.2 Zone Commands 14

zones/query 14

zones/<*zoneId*>/query/voltage 14

zone/<*zoneId*>/query/temperature 14

zone/<*zoneId*>/cmd/trigger/open 14

zone/<*zoneId*>/cmd/toggle/bypass?code=<*accessCode*> 14

4.3 General Commands 15

query/all 15

get/status 15

get/topology 15

refresh/topology 15

4.4 System Commands 15
system/get/status 15

4.5 Keypad Commands 16
keypad/<*kpId*>/cmd/press/funcKey/<*fkId*> 16
keypad/<*kpId*>/query/temperature 16
keypad/<*kpId*>/get/status 16

4.6 Output Commands 17
outputs/query 17
output/<*outputId*>/get/status 17
output/<*outputId*>/cmd/on?offTimerSeconds=<*seconds*> 17
output/<*outputId*>/cmd/off 17

4.7 Audio Commands 18
audio/zone/<*audioZone*>/source/<*audioSource*>/cmd/<*audioCommand*>?value=<*audioValue*> 18

4.8 Voice Announcement Commands 18
speak/word/<*wordId*> 18
speak/phrase/<*phraseId*> 18

4.9 Thermostat Commands 19
tstat/<*tstat_id*>/query 19
tstat/<*tstat_id*>/get/status 19
tstat/<*tstat_id*>/cmd/<*cmd_id*>?value=*value* 19

Revision History

Date/Firmware	Type	Change	Description
2012/03/16 **3.2.0**	DOC	NONE	
2011/12/02 **3.1.14**	WSDL	FIX	Fixed descriptions for display text on keypads Fixed some incorrect REST definitions Add Trigger Zone
2011/11/21 **3.1.13**	WSDL	NEW	Support for ELK Thermostats (sections 3.7 and 4.9) elkobjs.xsd udielkws1.wsdl **New Events**: Zone Event/Temperature (3/56) Keypad Event/Temperature (4/113) **New Services:** GetAreaStatus QueryZoneStatus GetZoneStatus QueryZoneTemperature QueryKeypadTemperature GetKeypadStatus QueryOutputs GetOutputStatus ThermostatCmd QueryThermostat GetThermostatStatus QueryAll GetAllStatus GetSystemStatus
2011/11/14 – **3.1.12**	WSDL	NEW	Initial

1. Introduction

ISY is an award winning platform for automation and energy management. With the introduction of ELK Integration Module (21090), all ELK events can now be acted upon as well as published to clients. Furthermore, developers can use Web Services/REST interface for direct communications with the ELK system.

2. Getting Started

Communications and event infrastructure follow the same paradigm as those defined in ISY's WSDK Developer's guide. Additional events, Web Services, and REST interface are defined herein. 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 ELK Integration module (21090) installed on ISY, please send an email to sales@universal-devices.com with your UUID (Help | About) or purchase the module through Help | Purchase Modules on the Admin Console.

Once you are successfully communicating with ISY and have ELK Module 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 Orchestrator. In the <serviceList> element, look for **UDIELKWebServices** 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>UDIELKWebServices</serviceType>
            <serviceId>
                  uuid:00:03:f4:03:65:96-UDIELKWebServices
            </serviceId>
            <SCPDURL>/elkServices.wsdl</SCPDURL>
            <controlURL>/security/elk</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 ELK web services

3. Get ISY Configuration (GetISYConfig Web Service or /rest/config) and ensure that ELK Integration module is installed (id = 21090). For more information on Configuration Resources and Modules, please consult ISY WSDK Developer's Guide
4. All ELK web services are defined in the WSDL returned by step 2. All objects are defined in **elkobjs.xsd** and are imported into the WSDL
5. If you are using SOAP UI, you can immediately communicate and issue ELK services to ISY

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

For offline tests, the WSDL file is: ***udielk1.ws***

3. ELK Events (control = "_19")

In addition to all the events published by ISY framework, ELK module has its own set of events that are specific to ELK Security System.

*All events are defined in ***elkobjs.xsd***.

3.1 Topology Changed (action = "1")

Topology is the complete configuration for ELK and thus, in the case of this event, it's best to use corresponding Web Services (GetTopology) or REST (/get/topology) to retrieve the topology.

3.2 Area Event (action = "2")

This event is published when something in a define Area changes. These include **Alarm** status changes, **Arming** up status changes, and **Armed** status changes.

EventInfo structure will be of the form:
<eventInfo>
    <**ae** …. />
</eventInfo>

Where **ae** is defined in:

Schema File	**elkobjs.xsd**
Namespace	**uelk**
Class	**ELKAreaEventInfo**

3.3 Zone Event (action = "3")

This event is published when something in a define Zone changes. These include **Logical** status changes, **Physical** status changes, and **Voltage** changed.

EventInfo structure will be of the form:
<eventInfo>
    <**ze** …. />
</eventInfo>

Where **ze** is defined in:

Schema File	**elkobjs.xsd**
Namespace	**uelk**
Class	**ELKZoneEventInfo**

3.4 Keypad Event (action = "4")

This event is published when something in a defined Keypad changes. These include **Access Code** status changes, **Key** status changes, and **LED** status changes.

EventInfo structure will be of the form:
<eventInfo>
    <**ke** …. />
</eventInfo>

Where **ke** is defined in:

Schema File	**elkobjs.xsd**
Namespace	**uelk**
Class	**ELKKeypadEventInfo**

3.5 Output Event (action = "5")

This event is published when something in a defined Output changes. These include **Output** status changes.

EventInfo structure will be of the form:
<eventInfo>
    <**oe** …. />
</eventInfo>

Where **oe** is defined in:

Schema File	**elkobjs.xsd**
Namespace	**uelk**
Class	**ELKOutputEventInfo**

3.6 System Event (action = "6")

This event is published in two cases: 1) ISY/ELK connection events and 2) Module enable/disable events.

EventInfo structure will be of the form:
<eventInfo>
    <**se** …. />
</eventInfo>

Where **se** is defined in:

Schema File	**elkobjs.xsd**
Namespace	**uelk**
Class	**ELKSystemEventInfo**

3.7 Thermostat Event (action = "7")

This event is published in the case of thermostat changes of state.

EventInfo structure will be of the form:
<eventInfo>
    <**te** …. />
</eventInfo>

Where **te** is defined in:

Schema File	**elkobjs.xsd**
Namespace	**uelk**
Class	**ELKThermostatEventInfo**

4. REST Interface

REST is an easy to use URL based command set which allows the developer to communicate and control ELK services through ISY.

All REST commands use HTTP GET method.

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

WSDL:isyelk:UDIDefaultResponse

URL Prefix: /rest/elk/

4.1 Area Commands

areas/query

Queries all areas and changes to states are published through event infrastructure (see section 3.2).

area/<*areaId*>/get/status

Retrieves the status of the given area.

areaId	Defined in **elkobjs.xsd:uelk:AreaIDType**

Response	**elkobjs.xsd:uelk:AreaResponseType**

area/<*areaId*>/cmd/bypass?code=<*accessCode*>

Bypasses all violated burglar alarms in the area given in areaId

areaId	Defined in **elkobjs.xsd:uelk:AreaIDType**
accessCode	Optional; Defined by **elkobjs.xsd:uelk:AccessCode**

area/<*areaId*>/cmd/unbypass?code=<*accessCode*>

Unbypasses all burglar alarms in the area given in areaId

areaId	Defined in **elkobjs.xsd:uelk:AreaIDType**
accessCode	Optional; Defined by **elkobjs.xsd:uelk:AccessCode**

area/<*areaId*>/cmd/display?text=<*eText*>&beep=<boolean> &offTimerSeconds=<*seconds*>

Displays text given in eText to all keypad in the given area. Optionally beep and turn off after offTimerSeconds.

areaId	Defined in **elkobjs.xsd:uelk:AreaIDType**
eText	URL escaped text. 2 Lines with 16 characters per line max. Use '^' to separate lines if less than 16 characters. e.g. text="Hello^World"
beep	Optional; True then beep when displaying the message
seconds	Optional; Number of seconds to display the message

area/<*areaId*>/cmd/display?id=<*notifyId*>&beep=<boolean> &offTimerSeconds=<*seconds*>

Displays formatted text to all keypad in the given area given a custom content ID. This uses the same custom content entries as are used for email notifications. Optionally beep and turn off after offTimerSeconds.

areaId	Defined in **elkobjs.xsd:uelk:AreaIDType**
notifyId	The id of the customized content entry
beep	Optional; True then beep when displaying the message
seconds	Optional; Number of seconds to display the message

area/<*areaId*>/cmd/arm?armType=<*elkArmType*> &code=<*accessCode*>

areaId	Defined in **elkobjs.xsd:uelk:AreaIDType**
elkArmType	Defined by **elkobjs.xsd:uelk:ArmType**
accessCode	Optional; Defined by **elkobjs.xsd:uelk:AccessCode**

area/<*areaId*>/cmd/disarm?code=<*accessCode*>

areaId	Defined in **elkobjs.xsd:uelk:AreaIDType**
accessCode	Optional; Defined by **elkobjs.xsd:uelk:AccessCode**

4.2 Zone Commands

zones/query
Queries all zones and changes to states are published through event infrastructure (see section 3.3).

zones/<*zoneId*>/query/voltage
Queries the zone given by zoneId and changes to states are published through event infrastructure (see section 3.3).
To query voltage for all zones, specify a *zoneId* of 0 (zero).

zoneId	Defined in **elkobjs.xsd:uelk:ZoneIDType**

zone/<*zoneId*>/query/temperature
Queries the thermostats for the given zone and changes to states are published through event infrastructure (see section 3.3).
To query temperature for all zones, specify a *zoneId* of 0 (zero).

zoneId	Defined in **elkobjs.xsd:uelk:ZoneIDType**

zone/<*zoneId*>/cmd/trigger/open
Momentarily triggers a zone to the physical state of *Open.* An error will occur if the zone is defined as normally open, or is currently open.

zoneId	Defined in **elkobjs.xsd:uelk:ZoneIDType**

zone/<*zoneId*>/cmd/toggle/bypass?code=<*accessCode*>
Toggles bypass for the zone given in zoneId

zoneId	Defined in **elkobjs.xsd:uelk:ZoneIDType**
accessCode	Optional; Defined by **elkobjs.xsd:uelk:AccessCode**

4.3 General Commands

query/all
Gets the status for all areas, zones, outputs, counters, etc. and publishes the state changes through event infrastructure (see section 3).

get/status
Returns the status of all of zones, areas, keypads, outputs, etc. configured in ELK.

Response	**elkobjs.xsd:uelk:ELKAllStatus**

get/topology
Returns the representation of all the zones, areas, keypads, outputs, etc. configured in ELK.

Response	**elkobjs.xsd:uelk:Topology**

refresh/topology
Causes ISY to recreate/refresh the topology. Returns the representation of all the zones, areas, keypads, outputs, etc. configured in ELK.

Response	**elkobjs.xsd:uelk:Topology**

4.4 System Commands

system/get/status
Returns the enabled and connected status.

Response	**elkobjs.xsd:uelk:SystemResponseType**

4.5 Keypad Commands

keypad/<*kpId*>/cmd/press/funcKey/<*fkId*>

Simulates pressing the fkId button on keypad given in kId

kpId	Keypad number (1-8)
fkId	Defined in **elkobjs.xsd:uelk:FunctionKeyType**

keypad/<*kpId*>/query/temperature

Queries the status of temperature sensor on a keypad and changes to states are published through event infrastructure (see section 3.3).
To query temperature for all keypads, specify a *kpId* of 0 (zero).

keypad/<*kpId*>/get/status

Retrieves the status of a keypad.
To get status for all keypads, specify a *kpId* of 0 (zero).

kpId	Keypad number (1-8)
Response	**elkobjs.xsd:uelk:KeypadResponseType**

Note: See Area Commands for displaying text on keypads

4.6 Output Commands

outputs/query

Queries all outputs and changes to states are published through event infrastructure (see section 3.5).

output/<*outputId*>/get/status

Returns the status of the given output
To get status for all outputs, specify an *outputId* of 0 (zero).

outputId	**elkobjs.xsd:uelk:OutputIDType**

Response	**elkobjs.xsd:uelk:OutputResponseType**

output/<*outputId*>/cmd/on?offTimerSeconds=<*seconds*>

Turns On an output defined by outputId. Optional offTimerSeconds can be defined such that the output is turned back off after the amount of seconds given in <seconds>.

outputId	**elkobjs.xsd:uelk:OutputIDType**
seconds	Optional; Interval after which the relay turns off

output/<*outputId*>/cmd/off

Turns Off an output defined by outputId

outputId	**elkobjs.xsd:uelk:OutputIDType**

4.7 Audio Commands

Audio commands allow communications with A/V equipment that have been configured and attached to ELK.

audio/zone/<*audioZone*>/source/<*audioSource*>/cmd/<*audioCommand*>?value=<*audioValue*>
Sends the command given in audioCommand to zone given in audioZone.

audioZone	Audio Zone number (1-18)
audioSource	Audio Source number (1-12)
audioCommand	Defined by **elkobjs.xsd:uelk:AudioCommandType**
audioValue	3-digit decimal Number, currently only used for Volume

4.8 Voice Announcement Commands

Voice Announcement commands cause predefined words or phrases to be spoken by the ELK security system.

speak/word/<*wordId*>
Instructs ELK to speak the word given by wordId.

wordId	Defined by **elkobjs.xsd:uelk:VoiceWordType**

speak/phrase/<*phraseId*>
Instructs ELK to speak the phrase given by phraseId.

phraseId	Defined by **elkobjs.xsd:uelk:VoicePhraseType**

4.9 Thermostat Commands

Thermostat commands impact a thermostat .

tstat/<*tstat_id*>/query

Instructs ELK query the thermostat given by tstat_id and state changes are published as events (see section 3.7).

tstat_id	Defined by **elkobjs.xsd:uelk:ThermostatIDType**

tstat/<*tstat_id*>/get/status

Retrieves the status of the given thermostat.

tstat_id	Defined by **elkobjs.xsd:uelk:ThermostatIDType**

Response	**elkobjs.xsd:uelk:ThermostatResponseType**

tstat/<*tstat_id*>/cmd/<*cmd_id*>?value=*value*

Instructs ELK to apply the value using command defined by cmd_id to thermostat defined by tstat_id.

tstat_id	Defined by **elkobjs.xsd:uelk:ThermostatIDType**
cmd_id	Defined by **elkobjs.xsd:uelk:ThermostatCommandType**
value	Depends on the command: Temperatures (such as setpoints): 1-99 Mode: **elkobjs.xsd:uelk:ThermostatModeState** Fan: **elkobjs.xsd:uelk:ThermostatFanState**