aws elemental

ELEMENTAL® STATMUX

API AND USER GUIDE

2.23.4.0 RELEASE

Mar 2022

TABLE OF CONTENTS

OVERVIEW

PURPOSE

This document is intended for system integrators and users of Elemental® Statmux. It outlines interfaces for machine and human control, configuration, and monitoring. Each API is defined in enough detail to explain how to use the system and how it can be integrated into larger workflow automation systems.

PRODUCT OVERVIEW

Elemental Statmux is a multiplexer which combines multiple Elemental Live outputs into a Multi-Program Transport Stream (MPTS). It can use advanced statistical multiplexing to dynamically allocate the MPTS bitrate among channels based on complexity.

Elemental Statmux can be controlled, configured and monitored through the following interfaces:

- Web browser via HTML
- Web Services REST interface

Using a web browser is the easiest way to control, configure, and monitor Elemental Statmux. This interface is used when a human is interacting with the server, or when no automation or integration with other systems is required. Elemental recommends Mozilla Firefox as the client web browser.

The REST-based interface supports all features of the web interface as well as automation features. More general information on REST-based interfaces is available online.

Finally, secure shell access allows the user to modify the system's configuration files, directory structure, and built-in tests. The secure shell interface is provided for users who need to modify the base behavior of the Elemental Statmux system or for diagnostics.

MPTS MULTIPLEXER

The Multi-Program Transport Stream multiplexer (MPTS mux) combines audio, video and data from multiple Events running on an Elemental Live node into a single MPEG-2 transport stream, output over UDP. MPTS muxes are managed from the MPTS Control page.

The MPTS mux takes its inputs from one or more Live Events. In order to be eligible for MPTS muxing, a Live Event must be configured with a single UDP/TS output group that has an â€˜MPTS Membershipâ€™. In order to participate in an MPTS mux running on an Elemental Statmux node, a setting of "Remote" is required. The output also must be attached to a stream that is using either CBR or Statmux as its Rate Control Mode.

When Live Events in an MPTS mux are configured with Statmux as their rate control mode, statistical multiplexing is used to allocate the total available MPTS bitrate among the video stream bitrates based on complexity. This is typically used for streams that are part of a fixed capacity transport mechanism. Bits are transferred dynamically from simple content to complex content, maximizing the overall visual quality of the output.

NOTIFICATION

Users can configure notifications to be sent if alerts are raised by running MPTS muxes. The user can be notified in the following ways:

- Email
- Web service callbacks - An HTTP POST will be performed to a URL that you provide, with information about the alert

TROUBLESHOOTING

Problems with Elemental Statmux may be diagnosed by viewing the log files available here: http://server_ip/logs.

For additional support, contact your Elemental support representative, or use the AWS Elemental Support Center in the AWS Console - http://amzn.to/AWSElementalSC -

WEB INTERFACE QUICK START GUIDE

WEB INTERFACE QUICK START GUIDE

Elemental Statmux includes a web interface to help you get started quickly. This page explains the basic steps to get started using the web interface and defines some common terms.

GETTING STARTED

Point a web browser at the Elemental Statmux web address: http://<ip address of server>

There are five base pages for the web interface

- MPTS Control: View an manage MPTS multiplexers and their channels
- Alerts: Access system and Live Event alerts
- Logs: Access system and Live Event alerts
- Settings: Modify Elemental Live settings
- Support: Documentation for the web interface, the REST interface, and the SNMP API

DEFINITION OF COMMON TERMS

- **Live Event:** a stream or streams to be broadcast in real-time from an Elemental Live node, specifies input source(s), output stream(s), output group(s), optionally may include a start/stop time and video effects to be applied to the output streams.
- **MPTS:** A Multi-Program Transport Stream multiplexer.
  How to set up an MPTS

DEFINITION OF COMMON ICONS

Icons are used to indicate both state and available actions. Many of these icons are not explicitly labeled (though if you hover over the icon, a tooltip will appear to indicate the icon's action).

 Show:
Shows additional information about an item

 Edit:
Indicates that the given object can be edited

 Play:
Starts an idle MPTS mux

 Stop:
Stops a running MPTS mux

 Delete:
Deletes the given item

 Working:
This icon indicates work in progress. It can appear while data is being loaded, or when starting and stopping an MPTS mux.

USING MPTS MULTIPLEXERS

MPTS muxes are managed from the MPTS Control page. From there you can create an MPTS, manage its channels, start or stop its output, and view a graph of performance statistics.

CREATING A NEW MPTS MUX

To create a new MPTS, select â€˜Create MPTSâ€™. At a minimum, provide a name, a transport stream ID, a transport stream bitrate, a video allocation bitrate, and a destination. The video allocation bitrate must be less than the transport stream bitrate, and the buffer between is intended for audio, data, and null packets. The destination can be a UDP or RTP address.

ADDING CHANNELS

Once an MPTS is created, click its name to view its channel listing. Live Events can be added as channels via the â€˜Add Channelâ€™ button. In order to be eligible for MPTS muxing, a Live Event must be configured with a single UDP/TS output group that has an â€˜MPTS Membershipâ€™ set. A setting of â€˜Localâ€™ allows the event to participate in an MPTS mux running on the Live node itself, while a setting of â€˜Remoteâ€™ configures it to communicate with a remote Elemental Statmux node. The output also must be attached to a stream that is using either CBR or Statmux as its Rate Control Mode. You must remove a Live Event from one MPTS before adding it to another.

When configuring a Live Event for MPTS membership, its UDP Settings define a set of destination values, including the *Destination*, the *Primary Complexity Transmit Destination*, and the *Primary Allocation Receipt Destination*. For â€˜Localâ€™ MPTS membership, the Elemental Live system sets the addresses for all destinations (these fields are hidden). For â€˜Remoteâ€™, multicast addresses must be added manually. The values set at the Live Event must correspond with values set at the MPTS. The arrangement of fields is intended to support a side-by-side view. â€˜Destinationsâ€™ go to â€˜Inputsâ€™, â€˜Transmitsâ€™ go to â€˜Receiptsâ€™, and â€˜Receiptsâ€™ go to â€˜Transmitsâ€™.

- The *Primary Destination* at the Live Event must match the *Primary Input* at the MPTS.
- The *Primary Complexity Transmit Destination* at the Live Event must match the *Primary Complexity Receipt Destination* at the MPTS.
- The *Primary Allocation Receipt Destination* at the Live Event must match the *Primary Allocation Transmit Destination* at the MPTS.
- All secondary destinations are optional, but must correspond with the same MPTS as the primary destinations.
- Corresponding IGMP Source and Virtual Sources per destination/input must match.

SETTING UP NETWORK REDUNDANCY FOR ELEMENTAL MPTSES

MPTS muxes and Live Events can communicate with full network redundancy. This means that all communication paths can be duplicated and sent through two different networks simultaneously. This provides protection from networking equipment failure and allows the flow of essential media and statmux algorithm data to flow error-free.

There are three streams of data for each channel of an MPTS, as follows:

- There is the flow of complexity estimates from the Event to the MPTS, which are used for statmux accounting and to weigh the channels against each other.
- There is a flow of bit rate allocations from the MPTS to the Event, which informs the encoder how many bits may be allocated to a section of content.
- There is the Single Program Transport Stream that flows from the Event to the MPTS containing the encoded media, ready for muxing into the final MPTS output.

Each of these streams of data is assigned a destination address. Typically, this is done with multicast addresses, but unicast addresses can also be used. Multicast addresses are required for functional Event and MPTS level failover, which is different from network redundancy.

The Live Event and MPTS provide the following address fields for each channel:

For SPTS data:

- Primary Destination
- Secondary Destination

For initial complexity estimates:

- Primary Complexity Transmit/Receipt Destination
- Secondary Complexity Transmit/Receipt Destination

For final allocations:

- Primary Allocation Transmit/Receipt Destination
- Secondary Allocation Transmit/Receipt Destination

When filling these out, only the primary addresses are required. If secondary destinations are specified, then the system will use two independent data flows, and implement an automatic switch between them as required.

To ensure that the two data flows are actually moving over separate networks it is strongly recommended that the "interface" parameter is used. This allows the data to be forced over a specific network interface (such as "eth1" or "eth2"). In this example eth1 and eth2 should be physically wired to two separate networks.

After configuring the Event, the MPTS should be similarly configured. The same multicast addresses should be used for each of the corresponding six fields, and interfaces should be specified to enforce that the data flows over the correct networks. It is not required that the same interface names be used on the MPTS and on the Event, but the corresponding interfaces should connect to the same network.

For example: Elemental Live Event "Primary Complexity Destination" is set to 230.100.100.100 on interface eth1, which is connected to network A. Elemental Statmux MPTS "Primary Complexity Destination" is set to 230.100.100.100 on interface eth3, which is connected to network A as well.

The Event and MPTS logs contain information about the percentage of dropped packets on the network, and the percentage of those drops that were recovered by using the data from the other network path.

PID ASSIGNMENT

When adding a channel, PID values for the MPTS output are automatically assigned to avoid collision, based on the existence of PID values in the Live Eventâ€™s configuration. For remote channels, all PID types are considered to be present. These values can then be edited, with validations in place to avoid collisions. Itâ€™s worth noting that, since PCR may be carried in another PID, assigning a PCR value in the MPTS is only relevant if it has been given an isolated PID value in the Live Event configuration.

STARTING OR STOPPING THE MPTS

An MPTS can be started or stopped from the MPTS index, or anywhere within the individual MPTS views. An MPTS must be stopped before being deleted.

USING THE ALERTS PAGE

The Alerts page is accessible from the main menu. It can also be accessed by clicking on an active alert in the upper right corner of any page.

Alerts can be generated for system level events including:

- **CPU Alert** - Cumulative CPU usage is too high
- **Disk Alert** - A disk partition is almost full
- **GPU Temperature Alert** - The GPU temperature is too high

The user can configure a notification email address or web callback for system alerts, as well as adjust the threshold for when these alerts are generated. Alerts will also trigger an SNMP Trap if a trap destination is configured in the SNMP Settings page.

Alerts can also be generated for an individual MPTS mux. Potential alerts include:

- **Stopped receiving network data on [URL]** - The UDP input is lost.
- **MPTS [input ID] dropped packets** - The muxer detects that it has dropped packets.
- **Failed to Build PID map due to insufficient output PIDs** - The PID mapping isnâ€™t sufficient for all of the incoming PIDs.
- **Protection Output Engaged** - When you have output listening turned on for a secondary Statmux and the Primary fails.
- **Failed Socket Open** - The UDP output writer couldn't open.
- **Failed Send** - The UDP output writer couldn't send.

USING THE SETTINGS PAGE

The settings page provides access to a variety of configuration options for Elemental Live.

GENERAL SETTINGS

The General Settings page allows for selecting the timezone for the Elemental Live system, and also provides an option for disabling the browser warning that appears on unsupported browsers. Elemental suggests setting the timezone before creating any Live Events or Schedules If the timezone is updated Elemental suggests restarting the service on the Elemental Live node, and recreating any Schedules. Note that disabling the browser warning only affects the current browsing session.

Periodic, automatic backups of the management database can also be configured from here. The available settings are the interval between performing the backups, how many database backups to keep, and the location on disk to store the backups. Entering a backup interval of every "0" minutes, disables the automatic backups.

To restore an automatic backup file called `elemental-db-backup_Statmux_2.19.2.0_2014-09-11_05-13-04.tar`, in the backup location `/home/elemental/database_backups`:

```
$ cd /opt/elemental_se/
$ sudo ./configure --restore-db-backup /home/elemental/database_backups/elemental-db-backup_Statmux_2.19
```

Additionally, settings for the Global Alert Notification are located on this page. The Global Alert Notification is a set of default notification settings that will be applied to any new alert that is created on the Elemental system.

NETWORK SETTINGS

The Network Settings page is divided into four sections, each accessible via a sub-tab on the left hand side. Please allow a few minutes for new settings to be applied to the system. In order to commit most changes, the "Save" button must be pressed. Restoring defaults will occur immediately.

CURRENT SETTINGS

The Current Settings sub-tab will display all information about the current network in a read only format. This includes hostname, DNS Servers, NTP Servers, IP address, netmask, and gateway for each ethernet device, and an output of the routing table.

HOSTNAME, DNS, & NTP

The Hostname, DNS & NTP sub-tab allows the changing of the hostname, the DNS name servers, and the NTP servers. Note that it is not possible to edit the name of an existing DNS name server or NTP server. The old name must be deleted and a new name added. NTP servers may be specified by name or IP address. DNS servers must be specified by IP address only.

NETWORK DEVICES

The Network Devices sub-tab allows for limited editing of network devices. Advanced properties such as bonds of multiple physical ports,or a Virtual Local Area Network (VLAN) devices are beyond the scope of this user interface. If you need help setting up one of these types of network device, please consult the appropriate Knowledge Base Article.

The "Edit" button next to each Network Device will bring up the "Edit a Network Device" dialog box with several available options:

- *Address Mode* - DHCP automatically assigns IP Address, Netmask, and Gateway. Static allows for specific configuration. None is also valid for eth.
- *Static Routes* - If checked, a table allowing creation of static routes using this network device be will exposed.

RESTORE DEFAULTS

The "Restore Defaults" button will replace any network devices with the system default.

MOUNT POINT SETTINGS

The Mount Point Settings page provides status information on active mount points and provides the ability to add new CIFS, NFS, or DAVFS mount points to the Elemental Statmux system. Mount points are limited to the /data/mnt directory.

Please allow a few minutes for the settings to be applied to the system.

FIREWALL SETTINGS

The Firewall Settings page provides access to the overall state of the firewall, and allows for the addition of new open TCP or UDP ports. When the firewall is on, you will see a list of all the open incoming ports that are managed by Elemental Statmux. There is a checkbox available to mark any open incoming ports for deletion, and there is a field below to add a new incoming TCP or UDP port. Incoming ports must be added one at a time.

Please allow a few minutes for the settings to be applied to the system.

SNMP SETTINGS

The SNMP Settings page provides access to the settings that allow or restrict SNMP access. There is an option to turn on SNMP traps for alerts and to set the port number that the manger receives the traps on. Please see SNMP Interface for more information.

Please allow a few minutes for the settings to be applied to the system.

AUTHENTICATION SETTINGS

The Authentication Settings page provides access to the settings that affect the authentication process. Authentication can only be enabled via the configure script. Once authentication is enabled, the authentication settings page controls the number of failed login attempts allowed and the length of time to ban a user after a failed login attempt, the session inactivity timeout, and whether to enable password expiration. See the Authentication page for more information.

WEB SERVICES REST INTERFACE

The Elemental Statmux system can be controlled through a REST interface over HTTP. A client program interacts with the server by sending HTTP GET, POST, PUT, or DELETE requests to resources on the server or server cluster. A wide range of available endpoints provide a simple interface to control and query all aspects of the Elemental system. Explore features of the REST API below.

REST BASICS

Representational state transfer (REST) is a style of software architecture for distributed systems such as the World Wide Web.

HTTP HEADERS

All requests must include the HTTP "Accept" header to specify the media type of the server's response. Responses can be HTML (Accept: text/html) or XML (Accept: application/xml). Requests that include a data payload (POST and PUT), must also include the HTTP "Content-Type" header to specify the media type of the data; Elemental supports only XML (Content-Type: application/xml). Additional headers are required when authentication is enabled on the server.

API VERSIONS

When submitting REST requests manually or from within an automation system, it is recommended to use an API version prefix for all endpoints. The API version prefix allows you to specify which API version the server should use to interpret your data. For example, POST http://<server_ip>/api/v2.19.2.0/mpts will send a request to the /mpts endpoint, and the server will interpret the data as compatible with Elemental API version 2.19.2.0. Although it is recommended that the API version prefix is included in all REST endpoints, omitting the prefix will assume the most current up-to-date API version: POST http://<server_ip>/api/mpts. Responses from the server will always be formed according to the current API version.

SIMPLE EXAMPLES

In all the following examples, replace server_ip with the IP address or DNS name of your Elemental server. To request a list of MPTS from the server, you can use cURL or a similar utility:

```
curl -H "Accept: application/xml" http://<server_ip>/api/mpts
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<mpts_list>
  <mpts>
    <id>1</id>
    <node_id>1</node_id>
    <additional_system_latency>0</additional_system_latency>
    <allocation_message_priority>primary</allocation_message_priority>
    <bitrate>38800000</bitrate>
```

```
    <name>MPTS of the Century</name>
    <output_listening>false</output_listening>
    <output_listening_interval>500</output_listening_interval>
    <pat_interval>100</pat_interval>
    <transport_stream_id>1</transport_stream_id>
    <buffer_msec>1000</buffer_msec>
    <video_allocation>35000000</video_allocation>
    <mpts_members>
      <mpts_member>
        ...
      </mpts_member>
    </mpts_members>
    <destination>
      <uri>udp://10.0.0.24:5003</uri>
    </destination>
  </mpts>
</mpts_list>
```

Adding or updating resources is accomplished by issuing an HTTP POST or PUT command with the body containing XML describing the resource. The client application must set the HTTP "Content-Type" header to: Content-Type: application/xml.

The xml contained in the file can also be entered inline after the -d option.

CLEAN XML

The XML that is returned by the server from a GET request is not in the correct format for creating new objects. The GET XML contains <id> tags to uniquely specify the object and any sub-objects, and it may also contain status information that will not be accepted by the server in a POST command. Being able to query the server for XML that is in a valid format for POSTing to create new objects is very useful -- it can be used to duplicate MPTS, or to slightly modify MPTS, MPTS Profiles, or Presets. Therefore, the Elemental Statmux REST interface offers a way to get 'clean' XML that is acceptable for creating new objects.

As an example, the following command gets the clean XML for MPTS 1. Simply make the regular GET request and add an extra parameter clean=true at the end.

```
curl -H "Accept: application/xml" http://<server_ip>/api/mpts/1?clean=true
```

This XML can be saved to a file and then POSTed back to the same server or another server to create an identical MPTS, or the file may be edited to make any necessary adjustments.

ERRORS AND WARNINGS

Validation errors when submitting an object are returned in the response XML.

MPTS MULTIPLEXERS

The MPTS API allows the user to start and stop an MPTS.

URL	METHOD	PARAMETERS	RETURNS	DESCRIPTION
/mpts	GET		List of MPTS XML	Get XML for every MPTS
/mpts/<mpts_id>	GET		MPTS XML	Get XML for a single MPTS
/mpts	POST	MPTS Parameters	MPTS XML of newly-created MPTS, or errors	Create a new MPTS
/mpts/<mpts_id>	PUT	MPTS Parameters	MPTS XML of updated MPTS, or errors	Update an existing MPTS
/mpts/<mpts_id>	DELETE		Successful response, or errors	Delete an existing MPTS
/mpts/<mpts_id>/mux	POST		Successful response, or errors	Start an MPTS

URL	METHOD	PARAMETERS	RETURNS	DESCRIPTION
/mpts/<mpts_id>/mux	DELETE		Successful response, or errors	Stop an MPTS
/mpts/statuses	GET		MPTS statuses	Get the status of every MPTS
/mpts/<mpts_id>/status	GET		MPTS status	Get the status of single MPTS
/mpts/<mpts_id>/stats	GET		MPTS bitrate stats	Get bitrate information for an MPTS
/mpts/<mpts_id>/mpts_members	GET		List of MPTS Member XML	Get all member channels of an MPTS
/mpts/<mpts_id>/mpts_members/:id	GET		MPTS Member XML	Get a single member channel of an MPTS
/mpts/<mpts_id>/mpts_members	POST	MPTS Member Parameters	MPTS Member XML of newly-created channel, or errors	Add a new channel member to an MPTS
/mpts/<mpts_id>/mpts_members/<mpts_member_id>	PUT	MPTS Member Parameters	MPTS Member XML of updated channel, or errors	Update an existing channel member of an MPTS
/mpts/<mpts_id>/mpts_members/<mpts_member_id>	DELETE		Successful response, or errors	Remove an existing channel member of an MPTS

EXAMPLE XML: CREATE AN MPTS

```
<mpts>
  <bitrate>32000000</bitrate>
  <video_allocation>30000000</video_allocation>
  <fec_output_settings_id nil="true"/>
  <name>Simple MPTS from XML</name>
  <node_id nil="true"/>
  <output_listening>false</output_listening>
  <pat_interval>100</pat_interval>
  <transport_stream_id nil="true"/>
  <buffer_msec>1000</buffer_msec>
  <destination>
    <uri>udp://localhost:5555</uri>
  </destination>
  <secondary_destination>
    <uri>udp://localhost:5556</uri>
  </secondary_destination>
</mpts>
```

SETTINGS

Settings provides information on overall system settings. The REST interface can only query information about the settings. Any settings updates must be made via the UI.

URL	METHOD	RETURNS	DESCRIPTION
/settings	GET	Timezone, Network Settings, Firewall Settings, Mount Point Settings, Authentication Settings, Sequencer Settings, (Cluster Settings if part of a cluster)	Retrieves information about the current system settings. This XML is in a format that is accepted by the configure script (-i <filename>). This can be used to configure many identical boxes.
/settings?cluster=true	GET	Timezone, Network Settings, Firewall Settings, Mount Point Settings, Authentication Settings, Sequencer Settings, Cluster Settings	Retrieves information about the current system settings including example cluster settings configured such that the current server is the master node. This can be used to help configure Slave nodes after the Master node has been configured.

URL	METHOD	RETURNS	DESCRIPTION
/settings/network	GET	Network Settings	Retrieves information about the current network settings. Other Elemental Server units can communicate on the interface marked <management_interface> in a clustered environment.
/settings/mount_points	GET	Mount Point Settings	Retrieves information about the current mount point settings.
/settings/firewall	GET	Firewall Settings	Retrieves information about the current firewall settings.
/settings/snmp	GET	SNMP Settings	Retrieves information about the current SNMP settings.
/settings/authentication	GET	Authentication Settings	Retrieves information about the current authentication settings.
/settings/advanced	GET	Sequencer Parameters	Retrieves information about the current sequencer settings.
/settings/cluster	GET	Cluster Settings	Retrieves information about the current cluster settings. Only available for clustered systems.
/settings/stop	POST	<stop></stop>	Enables a graceful shutdown of the Elemental service. Currently running MPTS will run to completion, but no new MPTS will be started. When all MPTS have completed the service will shut down.
/settings/start	POST	<start></start>	Sends a start command to the Elemental service. Used to restart the service after a /settings/stop command.
/settings/advanced	POST	Sequencer Parameters	Updates Sequencer settings. Requires a restart of the Service in order to take effect.

ALERTS AND MESSAGES

The alerts API provides information about current alert conditions on the system. Messages provide more information about the results of MPTS

URL	METHOD	PARAMETERS	RETURNS	DESCRIPTION
/alerts	GET	Pagination parameters, Filter parameters can be appended to the URL, eg: /alerts?filter=all	List of alerts	Active (or all if filter=all) alerts for the system.
/messages	GET	Pagination parameters, Filter parameters can be appended to the URL, eg: /messages?filter=Error	List of messages	Messages can be Errors, Warnings, or Audit messages. They have a code and a text message, and are associated with a particular MPTS. See Codes for common messages codes.

AUTHENTICATION AND REST

When authentication is enabled on the Elemental Statmux system, additional information must be sent with the REST command in order to properly authenticate the request. The following additional headers must be set: X-Auth-User, X-Auth-Expires, X-Auth-Key.

The **X-Auth-User** header contains the login of the user to authenticate.

The **X-Auth-Expires** header contains the Unix timestamp (in UTC) that indicates the time after which the server will no longer accept the request as valid. For security purposes, Elemental recommends that this value should be ~30 seconds in the future.

The **X-Auth-Key** header should be constructed using the following algorithm:

```
md5(api_key + md5(url + X-Auth-User + api_key + X-Auth-Expires))
```

Each parameter in this expression should be entered as a string, and the '+' operator indicates string concatenation without any delimiters. The **api_key** parameter is the user's secret API key that can be retrieved on the User Profile page. For security, it is recommended that this key be reset periodically. The **url** parameter is the path part of the request URL minus any query parameters ***and*** without any API version prefix.

For example, consider a GET request to https://<server_ip>/api/mpts/1?clean=true by the user 'admin' with the api_key '1acpJN7oEDn3BDDYhQ' that expires on June 1, 2011 UTC. In this case the url parameter is '/mpts/1' and the X-Auth-Expires value is '1306886400'. Thus the value of X-Auth-Key should be computed as follows:

```
md5('1acpJN7oEDn3BDDYhQ' + md5('/mpts/1'+'admin'+'1acpJN7oEDn3BDDYhQ'+'1306886400'))
=> md5('1acpJN7oEDn3BDDYhQ' + md5('/mpts/1admin1acpJN7oEDn3BDDYhQ1306886400'))
=> '180c88df8d0d4182385f6eb7e7045a42'
```

This is a single access request, it is not persisted. If another request needs to be made, the X-Auth-Key must be recalculated and all the headers must be set correctly.

AUTHCURL SCRIPTS

In order to help construct and set these headers correctly, two helper scripts (auth_curl.rb and auth_curl.pl) can be found in **/opt/elemental_se/web/public/authentication_scripts**. These scripts show how to construct and set the headers correctly using Ruby or Perl. In addition, they can be used outright to ease the use of setting these headers using cURL.

Using the same example from above, to send a GET request to '/mpts/1' using the user 'admin' with the api_key '1acpJN7oEDn3BDDYhQ', simply use the following command:

```
./auth_curl.[rb|pl] --login admin --api-key 1acpJN7oEDn3BDDYhQ \
-H 'Accept: application/xml' https://<server_ip>/api/mpts/1
```

The script will use an X-Auth-Expires header that is 30 seconds in the future, and it will calculate the X-Auth-Key header and set all the additional headers correctly. Any additional options beyond the --login and --api-key options will be passed to cURL. When using the scripts in this manner, it does not matter if the Ruby or Perl scripts are used as their function is identical.

POST and PUT requests can also be issued using the helper scripts. For these cases it is important to remember to include an appropriate HTTP "Content-Type" header, as well as specifying your xml data payload. Here is an example of this usage:

```
./auth_curl.[rb|pl] --login admin --api-key 1acpJN7oEDn3BDDYhQ \
  -X [POST|PUT] \
  -H 'Accept: application/xml' -H 'Content-Type: application/xml' \
  -d @filename https://<server_ip>/api/v2.19.2.0/mpts
```

Note: HTTPS is only supported by default on Elemental Cloud nodes, and must be manually configured for other products.

SNMP INTERFACE

The Elemental Statmux system can be monitored and controlled through Simple Network Management Protocol (SNMP). If configured to do so, the system will generate SNMP traps for certain events like Alerts or MPTS errors.

A user can interact with the system using a variety of network management systems. Elemental Statmux includes the Net-SNMP (http://www.net-snmp.org/) command-line tools to access the SNMP interface while logged into the system over SSH. Examples in this document are given using net-snmp commands.

SNMP BASICS

External access to the SNMP interface can be enabled in the Settings -> SNMP tab. This setting will open the SNMP port on the firewall. If the firewall is disabled, then external SNMP access will be enabled. The SNMP interface is always available for local requests from an SSH session.

The SNMP interface can be queried using SNMP Get and Get Next requests, along with an object identifier (OID). OIDs define a hierarchy of variables that can be returned; the root of the Elemental OID hierarchy is 1.3.6.1.4.1.37086. SNMP requests should use version 2c, and there is a read-only community called elemental_snmp that has access to the Elemental subtree as well as a large number of other SNMP variables provided by the Net-SNMP agent. There is a writable community called elemental_snmp_write that provides write access to the Elemental subtree. An example request to check the status of the elemental_se service is as follows:

```
snmpget -c elemental_snmp -v 2c localhost 1.3.6.1.4.1.37086.1.0
```

returns

```
SNMPv2-SMI::enterprises.37086.1.0 = INTEGER: 1
```

Elemental provides Management Information Bases (MIBs) that give descriptive names to OIDs and defines relationships between them. There are two MIBs included:

- http://<server_ip>/mib/ELEMENTAL_MIB.txt - Base MIB for all Elemental products

These MIBs are installed on the system by default, and can be used with the net-snmp tools to get the same value as the above example:

```
snmpget -c elemental_snmp -v2c -m ELEMENTAL-MIB localhost serviceStatus
```

returns

```
ELEMENTAL-MIB::serviceStatus.0 = INTEGER: 1
```

SNMP OPERATIONS

The following variables from the base ELEMENTAL-MIB can be Get or Set via SNMP:

VARIABLE	TYPE	GET VALUES	SET VALUES
ELEMENTAL-MIB::serviceStatus	Integer	0 if the elemental_se service is not running, 1 if the service is running	0 stops the elemental_se service. 1 starts the service, and 2 restarts the service
ELEMENTAL-MIB::firewallStatus	Integer	0 if the system's firewall is off, 1 if on	1 will load new firewall settings. Firewall settings are configured in the Elemental web interface.
ELEMENTAL-MIB::networkSettings	Integer	Will always return 1. Required for some network management systems	1 will load new network settings. Network settings are configured in the Elemental web interface.

VARIABLE	TYPE	GET VALUES	SET VALUES
ELEMENTAL-MIB::mountPoints	Integer	Number of user-mounted filesystems in /mnt	1 will load new mount settings. Filesystem mount settings are configured in the Elemental web interface.
ELEMENTAL-MIB::version	String	Product version	
ELEMENTAL-MIB::httpdStatus	Integer	0 if the httpd service is not running, 1 if the service is running	0 stops the httpd service. 1 starts the service, and 2 restarts the service
ELEMENTAL-MIB::databaseBackup	Integer	1 if writes (starting backups) are allowed. 0 if writes are not allowed	1 starts a database backup, any other value in a SET command is an error.

SNMP TRAPS

The Elemental Live system can generate SNMPv2 Traps when certain events occur. This functionality can be enabled in the Settings -> SNMP tab by filling in the host, port, and community of the management system that will be receiving SNMP traps.

SNMP Traps are generated for the following events:

NOTIFICATION	EVENT	CONTENTS
ELEMENTAL-MIB::alert	Any alert generated by the system	ELEMENTAL-MIB::alertSet: 1 if the alert is being set, 0 if the alert is being cleared ELEMENTAL-MIB::alertMessage: Message describing the alert that was set or cleared ELEMENTAL-MIB::alertCompleteNotes: Complete notes for the alert that was set or cleared ELEMENTAL-MIB::alertNodeId: The numerical ID of the node generating the alert. ELEMENTAL-MIB::alertRunnableId: The numerical ID of the Job, Live Event, or Channel generating the alert, if applicable. ELEMENTAL-MIB::alertCode: The numerical code of the alert, if applicable. ELEMENTAL-MIB::alertSeverity: The severity of the alert, if applicable. ELEMENTAL-MIB::alertNodeHostname: The hostname of the node generating the alert, if applicable. ELEMENTAL-MIB::alertRunnableType: The type of runnable object generating the alert, if applicable. ELEMENTAL-MIB::alertRunnableName: The name of the Job, Live Event, or Channel generating the alert, if applicable.

AUTHENTICATION

The Elemental Statmux system can be enabled to require user authentication to access the UI and REST interface. Users can be configured to have a variety of different levels of access to the system, from read-only access to full access.

CONFIGURING AUTHENTICATION

Authentication can only be enabled by running the configure script with a special flag. Running the configure script in this mode will not affect any system settings besides authentication settings.

```
cd /opt/elemental_se

sudo ./configure --config-auth
```

This will launch the Authentication Configuration script. This script can be used to enable or disable authentication, and to update the admin user's information. When enabling authentication, the script will ask for the desired admin login, email and password, and create the admin user. The admin user has full access to the entire Elemental Statmux system, including User and Role management. If authentication is already enabled, running the script can be used to update the admin user's information, including the admin user's password, or to create new admin users.

Once authentication is enabled, a variety of authentication-specific settings will be available via the Authentication Settings page.

- The **Number of failed login attempts allowed** field specifies the number of login attempts allowed for a single user login before triggering a login timeout for that user login. This allows the Elemental Statmux system to protect against brute-force attacks. Setting this value to 0 will disable brute-force protection.
- The **Length of time to ban user after failed login attempt** specifies the login timeout length for a user that has triggered the maximum number of login attempts. Setting this value to 0 will enact a permanent ban for that user and is not recommended.
- If a user is inactive for the number of minutes specified in the **Inactivity timeout** field, then the user will be automatically logged out of the system. Setting this value to 0 disables this feature.
- Passwords can be set to automatically expire after some length of time, after which the user will be asked to reset their password. Checking **Enable Password Expiration** enables this feature.
- If password expiration is enabled, the **Passwords Expire After** field specifies the number of days between password resets. Note that this value applies to each user individually, and is calculated from the time the user last reset their password.

MANAGING ROLES

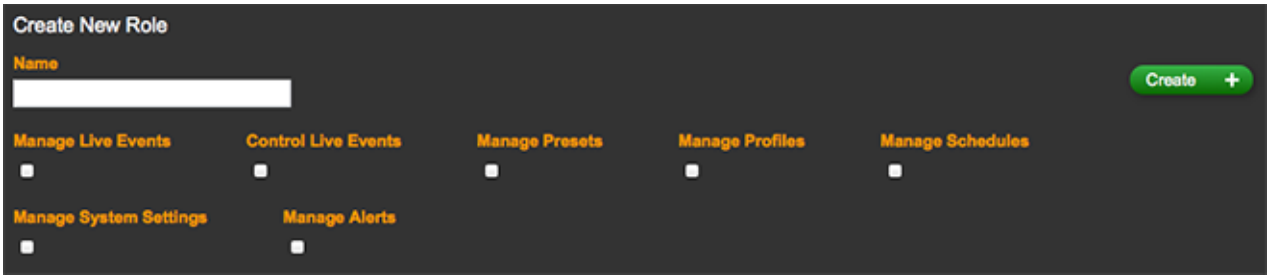
A user is assigned a specific role that defines the set of actions that user can perform. The Roles page can be found in the dropdown menu under Settings, and displays a list of existing roles, the number of users assigned to each role, and the full list of actions that role allows or disallows.

The Elemental Statmux system comes with a set of predefined Roles:

- **Admin**: The Admin role has access to the entire Elemental Statmux system
- **Viewer**: The Viewer role has read-only access to the Elemental Statmux system

CREATING NEW ROLES

In order to facilitate creating users that share a specific set of permissions, custom Roles may be created. Only admin users can create or edit roles. Roles are created by specifying what actions the role is allowed to access. Actions are grouped into a few large categories.



MANAGING USERS

The Admin user can create and manage users on the Users page, which can be found in the dropdown menu under Settings.

CREATING NEW USERS

To create a user, the admin user must fill out the Login, Password and Password Confirmation fields, as well as select the user's Role. The Password Expires field allows a user to be created with a password that will automatically expire after a set period of time. The Force Password Reset checkbox will force the user to reset their password the first time they login.

Admin users may also edit existing users, as well as reset their API keys, deactivate their access, and delete them entirely. Editing a user and checking the Force Password Reset will force that user to reset their password the next time they login. A deactivated user may be reactivated by editing the user and selecting any option besides Expired under the Password Expires dropdown.

USER PROFILE

Each logged-in user has access to their User Profile page, which can be found in the dropdown menu under Settings. The User Profile page displays the user's login, role, and API key (which is used for REST Authentication). The user may edit their email, reset their password, and update their API key from this page as well. In addition, a full list of the actions they may and may not perform is displayed.

AUTHENTICATION AND REST

Information on how to use the REST interface with authentication enabled can be found here.