ELEMENTAL® DELTA

API AND USER GUIDE

1.6.1 RELEASE

OCTOBER 5, 2015

TABLE OF CONTENTS

INTRODUCTION

This document is intended for system integrators and users of Elemental® Delta. 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.

OVERVIEW

Elemental Delta is a video delivery platform designed to optimize the management, monetization and distribution of video across internal and external IP networks. The platform provides a complete solution for time-shifted TV and just-in-time video packaging while enabling real-time content delivery with advanced levels of personalization, customization and control.

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

- Web browser via HTML
- Web Services REST interface
- SNMP interface

Using a web browser is the easiest way to control, configure, and monitor Elemental Delta. 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 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.

The SNMP interface allows basic monitoring and control of the Elemental Delta system. It allows a management system to query the state of the service and content it manages.

Secure shell access allows the user to access the system's configuration files and directory structure. The secure shell interface is provided for users who need to modify the base behavior of the Elemental Delta system or for diagnostics.

DEFINITION OF COMMON TERMS

- **Content:** Content defines the source input received by the system. Content can be either Linear (as sourced from products such as Elemental Live) or VOD (Video On Demand, as sourced from products such as Elemental Server). Content is received by the system through Input Filters.
- **Input Filters:** Input Filters define ingest points for content received by the system. They can control storage locations for content, retention windows, and content types.
- **Output Filters:** Output Filters allow the operator to manipulate content before being served to a requesting end-user. Output Filters can be chained together to create more complicated workflows. For example, an

operator can add an Ad Removal filter after a Live to VOD filter to create an endpoint which delivers a completed program without advertisements as a VOD asset to the end user.

- **Output Templates:** Output Templates are sets of Output Filters that can be applied to new Content. Input Filters can be configured to apply a particular Output Template to incoming Content.
- **Endpoints:** Endpoints define access points to Output Filters defined in Elemental Delta. When an Output Filter is configured as an Endpoint, it provides a URL for client devices to connect to. Elemental Delta will provide a unique URL for each Endpoint, but the user may also enter a custom URL. This custom URL can be applied to multiple Endpoints in the system. A common use case is to have unique User Agent filters at the end of several filter chains. When a client device makes a request to the Endpoint, Elemental Delta selects the filter chain that best matches the client device's User Agent HTTP header.

  When used in Output Templates, Endpoints can contain format identifiers based on input file names. The following format identifiers are available:

  - fn - Copies input file name into Endpoint URL
  - ex - Copies input file extension into Endpoint URL
- **Stream Sets:** Stream Sets allow the operator to assign certain input video streams, audio streams or subtitles tracks to outputs in a packaging filter. For example, when incoming MSS content has two audio languages, the operator can create a separate HDS packaging filter for each language and assign the appropriate audio stream to each video stream in the packaging filter. When a subtitles track in a supported input format is added to a stream set, Elemental Delta will repackage it on the fly to the right output format.

TYPES OF INPUT FILTERS

UDP INPUT FILTERS

A UDP Input Filter allows adaptive bitrate (ABR) content to be delivered to the Elemental Delta system as single-program MPEG Transport Streams over unicast or multicast UDP. The streams may have RTP headers. Content is stored on the filesystem as specified by the user. Each UDP Input Filter generates a corresponding TS Content entry in the Content page. One or more just-in-time packaging output filters should be applied to this content for delivery to end users.

WATCH FOLDER INPUT FILTERS

A Watch Folder Input Filter defines a directory accessible to Elemental Delta. The directory is monitored by the system for new input media files. When a new set of content is placed in a folder associated with a Watch Folder Input Filter, Elemental Delta will ingest the content and apply the associated Output Template. The directory can either be on the Elemental Delta system's disk drives or an external storage system mounted in the Settings -> Mount Points page. Watch Folder Input Filters can be used for VOD content.

WEBDAV INPUT FILTERS

WebDAV Input Filters allow content to be pushed to the Elemental Delta system over HTTP WebDAV (http://en.wikipedia.org/wiki/WebDAV). To define a WebDAV Input Filter, the operator configures a relative path, a set of user credentials, and a disk location to store incoming content. The operator should provide an upstream encoder or packager with the Absolute Path and the Username and Password of the selected WebDAV user. The operator can also apply an Output Template to new content.

WebDAV Input Filters can be used for Live content, or VOD content by selecting the "VOD Content" checkbox. The operator can define how long Elemental Delta should retain the content on disk by modifying the Content Window setting. The following options are available:

- Keep All Content - Content is retained by the system until the operator deletes it.
- Packager Controlled - The upstream packager can limit the content window by calling HTTP DELETE on old content segments.
- Keep Time Specified - Allows the operator to specify the age when content should be deleted, in either Seconds, Minutes, Hours, or Days.

MSS INPUT FILTERS

MSS Input Filters allow Microsoft Smooth Streaming content to be pushed to the Elemental Delta system over HTTP. To define an MSS input filter, the operator configures a relative path and a disk location to store incoming content. The operator should provide an upstream encoder or packager with the Absolute Path to deliver content. The operator can also apply an Output Template to new content.

MSS Input Filters are used for Live content. VOD content in MSS format is ingested via watch folder input filters. The operator can define how long Elemental Delta should retain live content on disk by modifying the Content Window setting. The following options are available:

- Keep All Content - Content is retained by the system until the operator deletes it.
- Keep Time Specified - Allows the operator to specify the age when content should be deleted, in either Seconds, Minutes, Hours, or Days.

REMOTE INPUT FILTERS

Remote Input Filters allow the processing and delivery of content that is originated by another web server. Elemental Delta will fetch content from the remote origin location and process it using the output filter chain. Segments that are retrieved from the remote origin are cahced in memory and disk caches configured in the Remote Input Filter for subsequent requests.

Only HLS content is supported for remote content. The operator adds available remote content via the "Add Content" button on a Remote Input Filter, or via the REST remote_input_contents API. When content is added, Elemental Delta will fetch the manifest files and enough content segments to analyze the stream for output processing.

Output filters can be applied to remote content, however Live to VOD, VOD Clip, Time Delay, MP4 Package, TS Package, and File Copy output filters are not valid output filters for remote content.

CONTENT TYPES

Elemental Delta supports the following content types for ingest and storage:

- HTTP Live Streaming (HLS) content (as defined here: http://tools.ietf.org/html/draft-pantos-http-live-streaming-11). This supports media delivery to Apple iOS devices, many Android devices, and several set-top boxes. Subtitles are supported through the WebVTT W3C standard (described here: http://dev.w3.org/html5/webvtt/)
- MPEG4 (MP4) files with H.264 video and AAC audio. Multiple MP4 files may be grouped together for adaptive bitrate content. MP4 files are ingested via the Watch Folder input filter. Files are not ingested until a SMIL (.smil) file (referencing the .mp4 files) is present in the watch folder. Example SMIL file referencing two MP4 files at different bitrates:

```
<?xml version="1.0" encoding="utf-8"?>
<smil>
  <head></head>
  <body>
  <switch>
    <video src="/var/lib/jenkins/jobs/PDFs/jobs/pdfs/workspace/web/publicvod_content_600k.mp4" />
    <video src="/var/lib/jenkins/jobs/PDFs/jobs/pdfs/workspace/web/publicvod_content_1.8m.mp4" />
  </switch>
  </body>
</smil>
```

- MPEG Transport Streams (TS). Multiple single program transport streams (SPTS) can be grouped together for ABR content. These streams should have presentation timestamps (PTS) aligned at keyframes for proper bitrate switching. They can have Encoder Boundary Point (EBP) markers to signal fragmentation points to Elemental Delta
- Microsoft Smooth Streaming (MSS) content for both Live and VOD applications. Subtitles are supported using the text-mode TTML format (described here: http://www.w3.org/TR/ttaf1-dfxp/).

Through the use of packaging filters, Elemental Delta can deliver DASH-ISO, HLS content, MSS content, MPEG-TS, MP4, and Adobe HTTP Dynamic Streaming (HDS) content to client devices. Through the use of the File Copy filter, MPEG-TS and MP4 files packaged by Delta can be stored to a filesystem.

ENCRYPTED CONTENT

Elemental Delta is capable of ingesting HLS content that has been encrypted using keys provided to the upstream packager by the Elemental Delta system. The user should configure their upstream packaging system to retrieve AES-128 encryption keys with an HTTP GET request as follows:

```
http://<server_ip>/keyserver?r=test&p=1404993276
```

The parameters supplied with this request are as follows:

- r: Resource ID. This is a unique identifier for a particular piece of content.
- p: Position. Index into the set of keys for this Resource ID.

When using Elemental Server, Elemental Live, or Elemental Stream to package content, select the "Generic Keyprovider" option for HLS Encryption to interface with Elemental Delta.

When encrpyted content is ingested, an indicator will be shown in the content window. The content will be stored in encrypted form, and will be decrypted before applying any output filters for delivery. To allow for end-to-end encrypted delivery, the operator should add an appropriate encryption output filter to the content.

TYPES OF OUTPUT FILTERS

GENERAL FILTERS

PASSTHROUGH

A passthrough filter will serve the content to the end user as it was formatted by the upstream packager. Passthrough is supported for HLS and MP4 content. Note that if the input filter’s content window is smaller than the packager’s, playback may be disrupted.

AD REMOVAL

Ad Removal filters can be attached to VOD content to remove ads. Input streams must be decorated with #EXT-CUE-OUT and #EXT-CUE-IN ad markers.

AD REPLACEMENT

Ad replace filters replaces the main content stream with the specified advertising content during ad avails. Input streams must be decorated with #EXT-CUE-OUT and #EXT-CUE-IN ad markers.

AKAMAI G2O AUTHENTICATION

This filter protects requests using Akamai Signature Header Authentication. Elemental Delta will only serve content from this filter when requested from an Akamai edge server with the correct authentication headers.

BLACKOUT

Blackout filters are used to select some content to be displayed when the program needs to be blacked out. The blackout functionality can be enabled/disabled (i.e. the replacement content or the original is displayed) upon an API call, or through the GUI.

BITRATE SELECTOR

Bitrate filters allow the operator to serve only certain video bitrates from a set of adaptive bitrate streams. When attached to content, the operator can enable and disable individual bitrates by toggling the selected bitrates. When used in an Output Template, the operator can specify minimum and maximum bitrates to serve. Five megabits can be entered as 5000000 or 5m. Five hundred kilobits can be entered as 500000 or 0.5m. When the template is applied to new content, only video bitrates in this range will be selected.

CACHE SETTINGS

Allows override of default max-age headers for all endpoints in this filter chain.

CISCO URL SIGNING

Adds URL signatures compatible with Cisco Internet Streamer CDS to bitrate playlist URLs for HLS outputs. This filter only adds signatures, it does not perform validation.

FILE COPY

When placed on file-oriented content (such as MP4) or after a file-oriented packaging filter (such as MP4 Package or MPEG-TS Package), the File Copy filter will copy the file(s) to a directory of your choosing. If a directory name (a name ending in a '/') is specified (in the path field) then the filter id will be used as the base name for files. If, on the other hand, a name is given after the '/' then that will become the base name for files. Subdirectories will be created if needed. See the MPEG-TS and MP4 Package filters for naming conventions for ABR content.

LIVE TO VOD

Live to VOD filters are used to turn a linear stream into a VOD asset. The start and end times are given to make a clip out of the linear stream. When a Content Window is specified in the input filter, content between the start and end times will be retained on disk when the input Content Window is passed. Live to VOD filters must be created before the content window expires in order for them to be retained. When the Content Window is set to Packager Controlled, the upstream packager must be configured to retain content to fit the Live to VOD window.

TIME DELAY

A time delay filter introduces a buffering delay between the incoming content and its play back. Input filter content window should be set larger than the desired time delay.

USER AGENT

User agent filters are used to select what user agents are considered to apply to the following elements in the filtering chain. For example, a user agent filter can be inserted before an ad replace filter, so that you can select different commercials when the same channel is viewed from a STB or a mobile device.

VOD CLIP

VOD Clip filters can clip portions of a content stream and deliver them as VOD streams. The start and end times are given relative to the beginning of the content.

PACKAGING FILTERS

DASH-ISO PACKAGE

Packages content for the DASH-ISO ABR streaming protocol. Can be applied to any content type. Endpoints can be accessed at http://<server ip>/out/i/<endpoint id>.mpd

HDS PACKAGE

Packages content for Adobe HTTP Dynamic Streaming. Can be applied to any content type. HDS package filter endpoints can be accessed at http://<server ip>/out/i/<endpoint id>.f4m

HLS PACKAGE

Packages content to Apple HTTP Live Streaming. Can be applied to any content type. Endpoints can be accessed at http://<server ip>/out/i/<endpoint id>.m3u8

MP4 PACKAGE

Packages content in the MP4 (file) format. Can be applied to any content type. Works with VOD content or downstream of a Live To VOD filter on live content. If multiple bitrates are selected you'll get multiple .mp4 files. Names will have stream index appended in that case e.g. moonwalk_1.mp4, moonwalk_2.mp4. Use File Copy filter to save the files to the filesystem. The files are avialable via HTTP progressive download on endpoints like http://<server ip>/out/i/<endpoint id>.mp4 SMIL manifest files may be accessed at http://<server ip>/out/i/<endpoint id>.smil

MPEG-TS PACKAGE

Packages content in the MPEG-TS (file) format. Can be applied to any content type. Works with VOD content or downstream of a Live To VOD filter on live content. If multiple bitrates are selected you'll get multiple .ts files. Names will have stream index appended in that case e.g. marsmission_1.ts, marsmission_2.ts. Endpoints can be accessed via HTTP progressive download at http://<server ip>/out/i/<endpoint id>.ts SMIL manifest files may be accessed at http://<server ip>/out/i/<endpoint id>.smil

MSS PACKAGE

Packages content for Microsoft Smooth Streaming players. Can be applied to any content type. MSS Package endpoints provide the manifest file at http://<server ip>/out/i/<endpoint id>.ism/Manifest

DRM FILTERS

FLASH ACCESS ENCRYPTION

Applies Adobe Flash Access encryption upon client request. Can apply Flash Access or pHDS encryption to HDS packaging filters, or Flash Access or pHLS to HLS content or HLS packaging filters.

HLS ENCRYPTION

HLS Encryption filters are used to apply a DRM/encryption algorithm to the content when requested by an end user. Elemental Delta is integrated with several key providers.

PLAYREADY ENCRYPTION

Playready filters are used to apply Playready DRM to MSS packaged content when requested by an end user. It must be chained after an MSS Package filter. Elemental Delta is integrated with several key providers.

COMMON ENCRYPTION

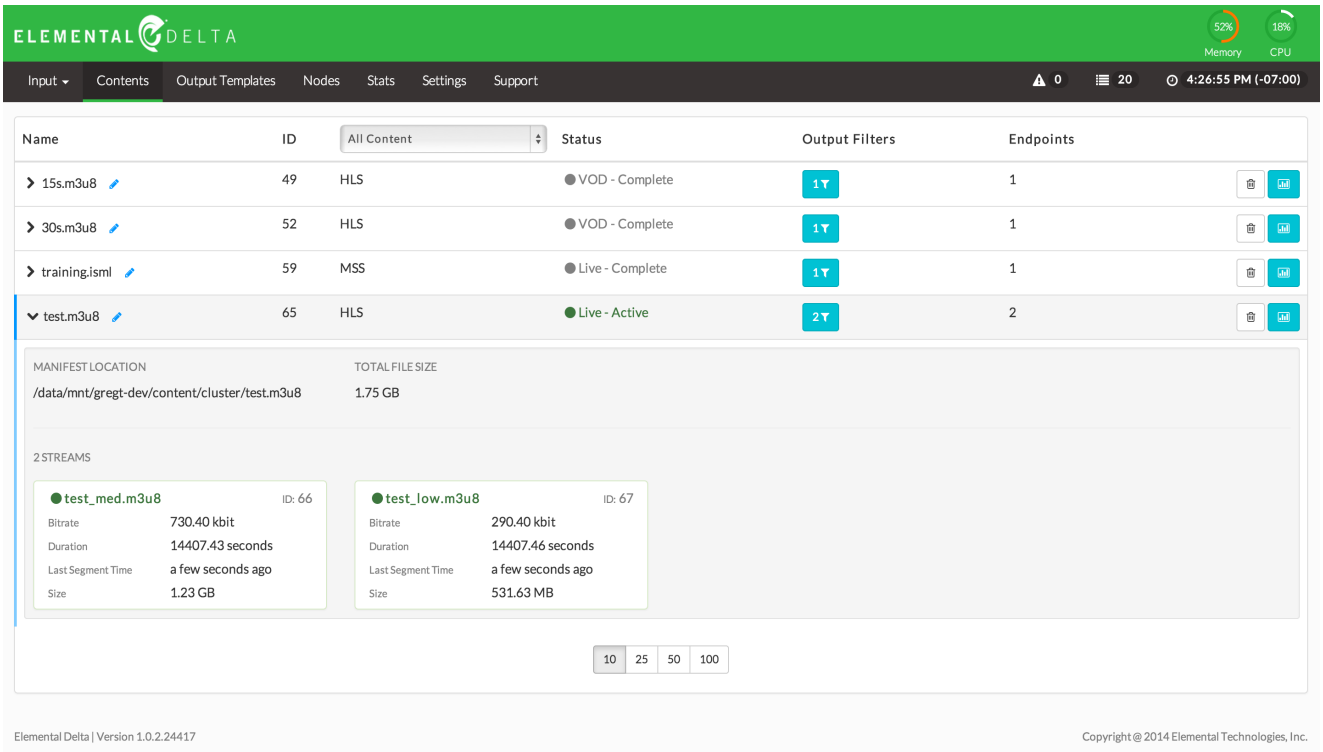
Common Encryption filters are used to apply CENC DRM/encryption algorithm to MPEG DASH-ISO output content.

TYPICAL STEPS FOR GETTING STARTED WITH ELEMENTAL DELTA

Point a web browser at the Elemental Delta web address at:

http://<ip address of server>:8080/

You should see a screen like this:



If this is the first time the system has been started, the Content list will be empty.

INGEST AND SERVE A STREAM OF LINEAR CONTENT THROUGH A BASIC PASSTHROUGH USING A WEBDAV INPUT FILTER

- Create a user with a name and password at [system]/input_users
- Create an Output Template at [system]/output_templates
  - Provide a name for your Output Template
  - Use the “Add Level 1 Filter” to add a Passthrough Filter
  - Set the Endpoint for the Filter to ‘true’. An endpoint with the incoming content's name will be automatically generated, so Output URL can be left blank.
  - Save the content configuration for this Filter.
- Create a WebDAV Input Filter at [system]/input_filters
  - Provide a name for the Input Filter.
  - Define a unique directory path. Writing a path that doesn’t yet exist will create the directories needed.
  - Select the Output Template you created earlier.
  - Define the system storage location and timing to keep as desired
- At your upstream system (example: Elemental Live) set your destination point to the Input Filter’s unique directory path, using the absolute URI. Include the username and password for the user you chose at the Input Filter. Publish this content.
- Return to the Elemental Delta system, and view the stream on the list of Content.
- You can control the distribution of this content by editing the Output Filters. To get to the configuration, select the edit (funnel) icon. Note that the configuration has been based on the Output Template, but is now separate from the template. Changes to this configuration will not update the template, and updates to the template will only influence future usage of the template.

- The system is now serving the content via the URL from the Filter Endpoint.

INGEST AND SERVE VOD CONTENT THROUGH A BASIC PASSTHROUGH USING A WATCH FOLDER INPUT

- Go to the mount point settings at: [system]/settings/mount_points.
- Specify the server share, mount point folder, username and password.
- Create an Output Template at [system]/output_templates
  - Provide a name for your Output Template.
  - Use the “Add Level 1 Filter” to add a Passthrough Filter.
  - Set the Endpoint for the Filter to 'true'.
  - Add a unique value for the Output URL. The full URL will display.
  - Save the content configuration for this Filter.
- Create a Watch Folder Input Filter at [system]/input_filters
  - Provide a name for the Input Filter.
  - Specify the path to the mounted folder.
  - Select the Output Template you created earlier.
- At your upstream system (example: Elemental Server) set your destination point to the mount point’s unique directory path.
- Return to the Elemental Delta system, and view the stream on the list of Content.
- You can control the distribution of this content by editing the Output Filters. To get to the configuration, select the edit (funnel) icon. Note that the configuration has been based on the Output Template, but is now separate from the template. Changes to this configuration will not update the template, and updates to the template will only influence future usage of the template.
- The system is now serving the content via the URL from the Filter Endpoint.

SERVE JUST-IN-TIME PACKAGED CONTENT TO HLS, MSS, AND HDS CLIENTS FROM UDP INPUTS

- Create an Output Template at [system]/output_templates
  - Provide a name for your Output Template.
  - Use the “Add Level 1 Filter” to add an HLS Package Filter.
  - Set the Endpoint for the Filter to 'true'.
  - Specify fn for Output URL. This will create endpoints from the incoming Content's name.
  - Save the content configuration for this Filter.
  - Repeat previous four steps for MSS Package and HDS Package filters
- Create a UDP Input Filter at [system]/input_filters
  - Provide a name for the Input Filter. This name will be applied to the package filters created in the output template, so it should be valid to appear in a URL.
  - Specify the multicast IP addresses and ports to ingest content from. Ranges can be specified with [brackets] and will be expanded to multiple entries upon hitting Add.
  - Optionally specify stream bitrates. Elemental Delta will automatically fill in a bitrate upon stream acquisition, however user-specified bitrates can be more accurate.
  - Select the type of segmentation markers used in your stream.
  - Select the Output Template you created earlier.
  - Specify a storage location on disk for incoming content.

  - Specify a content window to control how long content is retained on disk.

- Upon saving the Input Filter, a new Content object associated with the input filter will be created. Navigate to the Content page and expand the Content's details. If the upstream encoder is not yet publishing content, start it now. You should see Duration increasing until it matches the content window, and Size reflects how much disk space is being used by the content.
- To view details on endpoints, select the "Edit" button for the content. You should see the 3 Output Filters from the template. Clicking on the HLS Package filter will show the .m3u8 endpoint you can provide to an HLS player (such as an iPad).
- The system is now serving the content via the URLs from each Output Filter.

CONTENT STATISTICS

Elemental Delta tracks usage statistics at several levels. Clicking on the Stats button on the right of each Content row on the Contents page will show the statistics for that content. Clicking on any of the chart icons will generate a charts for the given time period. The operator can select a more fine-grained time period inside the chart's popup window.

Users can see total bytes sent to downstream clients, total HTTP requests made by downstream clients, and total errors generated for a given content item. This data is also available as a CSV file download, accessible from the statistics page.

Additionally, the same statistics are available for individual filters and their associated endpoints. This allows the operator a more detailed view of endpoint utilization by downstream users for each item of content.

PLAYBACK PREVIEW

The web user interface for Elemental Delta contains an embedded player for the operator to preview the content from various output filters. When editing an Output Filter which is configured as an endpoint in the Content details page, there is a link to the right of the URL which allows the operator to download the top-level manifest of that endpoint. To the right of that button, the Preview button will enable the embedded player in a modal popup window.

The preview player supports playback of HLS, HDS, and MSS content. Please note that it does not support all of the features offered by each streaming format. Additionally, if content is not available for the selected output filter, or the content is encrypted, the preview player will not play back content. This feature is intended to be used by the operator to check functionality of the system, but should not be distributed to end users for content playback.

USING THE SETTINGS PAGE

The settings page provides access to a variety of configuration options for Elemental Delta. This includes General Settings, Network Settings, Mount Point Settings, Firewall Settings, and SNMP Settings

GENERAL SETTINGS

The General Settings page lets you configure the timezone for the Elemental Delta system. If client players access Elemental Delta through a load balancer or other external hostname or IP address, configure that URL in the Public Endpoint field. That URL will be displayed in output filters and the preview player too. There are options here for configuring database backups and for managing default alert notifications.

NETWORK SETTINGS

The Network Settings page allows the operator to configure network interface cards, DNS settings, NTP settings, and advanced routing. Please allow a few minutes for new settings to be applied to the system. In order to commit most changes, the "Apply Changes" button must be pressed. Restoring defaults will occur immediately.

DNS AND NTP

The DNS and NTP sections allows the changing of 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.

After making any change on this page, the "Apply Changes" button must be clicked for the changes to take effect.

NETWORK DEVICES

The Network Devices section allows the creation of new network devices and editing of existing devices. A network device is equivalent to a physical Ethernet port on the chassis, a bond of multiple physical ports, or a Virtual Local Area Network (VLAN) device. VLAN devices add IEE 802.1q tagging to ethernet frames, and must be used with a compatible upstream switch. Ethernet devices can be added, deleted, or edited. The network device "eth0" can never be deleted. The "Edit" button next to each Network Device will bring up the "Edit a Network Device" dialog box.

To add a new network device, select the "Add Device" button and select the device type: bond, Ethernet, or VLAN. Only one device can be added through this dialogue at a time. Below are settings common for all types:

- *Management* - If checked, this network device will be used for communication between nodes within a cluster.
- *Description* - a description string for use by the user. No effect on system performance. Note that if a description is present during configuration, the system reads and uses that description.
- *Address Mode* - DHCP automatically assigns IP Address, Netmask, and Gateway. Static allows for specific configuration. None is also valid for bond and eth.
- *Static Routes* - Create static routes using this network device.

Below are settings unique to device types:

eth (ethN)

- *Device Name* - Specific name of device (eth1, eth2). The system will clarify if all eth devices are already configured.
- *Master Device* - If port bonding devices are configured, they will display here as options.

VLAN (ethN.M)

- *Parent* - Physical or bond device that this VLAN operates on.
- *VLAN ID* - An integer between 1-4094.

bond (bondN)

Bond ID - Must be an integer.

- Mode (for Port Bonding) - Select one of the following entries.
  - *0: Round Robin* - Sets a round-robin policy for fault tolerance and load balancing. Transmissions are received and sent out sequentially on each bonded slave interface beginning with the first one available.
  - *1: Active Backup* - Sets an active-backup policy for fault tolerance. Transmissions are received and sent out via the first available bonded slave interface. Another bonded slave interface is only used if the active bonded slave interface fails.
  - *2: Balanced XOR* - Sets an XOR (exclusive-or) policy for fault tolerance and load balancing. Using this method, the interface matches up the incoming request's MAC address with the MAC address for one of the slave NICs. Once this link is established, transmissions are sent out sequentially beginning with the first available interface.
  - *3: Broadcast* - Sets a broadcast policy for fault tolerance. All transmissions are sent on all slave interfaces.
  - *4: IEEE 803.ad Dynamic Link Aggregation* - Sets an IEEE 802.3ad dynamic link aggregation policy. Creates aggregation groups that share the same speed and duplex settings. Transmits and receives on all slaves in the active aggregator. Requires a switch that is 802.3ad compliant.
  - *5: Adaptive Transmit Load Balancing* - Sets a Transmit Load Balancing (TLB) policy for fault tolerance and load balancing. The outgoing traffic is distributed according to the current load on each slave interface.

Incoming traffic is received by the current slave. If the receiving slave fails, another slave takes over the MAC address of the failed slave.

- *6: Adaptive Load Balancing* - Sets an Active Load Balancing (ALB) policy for fault tolerance and load balancing. Includes transmit and receive load balancing for IPV4 traffic. Receive load balancing is achieved through ARP negotiation.

- *Link Mode* - Select whether to use MII or ARP link monitoring for all slaves in the bond. MII is generally preferred over ARP.
- *MII Monitoring Frequency* - Specifies the MII link monitoring frequency in milliseconds. This determines how often the link state of each slave is inspected for link failures. 100ms is a good starting point.
- *Use Carrier* - Used in conjunction with the MII Link Mode. If "Use Carrier" is selected then MII will use MII or ETHTOOL ioctls (less efficient, and uses deprecated kernel calling sequences), instead of netif_carrier_ok. Relies on the device driver to maintain link state.
- *Down Delay* - Specifies the time, in milliseconds, to wait before disabling a slave after a link failure has been detected. Only applies to the MII Link Mode, and should be a multiple of the MII Monitoring Frequency (will be rounded to nearest multiple). Defaults to 0.
- *Up Delay* - Specifies the time, in milliseconds, to wait before enabling a slave after a link recovery has been detected. Only applies to the MII Link Mode, and should be a multiple of the MII Monitoring Frequency (will be rounded to the nearest multiple). Defaults to 0.
- *ARP Interval* - Specifies the ARP link monitoring frequency in milliseconds. Periodically checks slave devices for traffic, generates regular interval traffic via ARP probes for ARP IP Target.
- *ARP IP Target* - Specifies the IP address to use for ARP probes in ARP Link Mode.

RESTORE DEFAULTS

The "Restore Defaults" button will replace any network devices with the system default, and remove all port bonding configurations.

MOUNT POINT SETTINGS

The Mount Point Settings page provides status information on the active mount points and provides the ability to add a new CIFS or NFS mount point to the Elemental Delta. 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 Delta. 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.

CLUSTERING DELTA NODES

Two Elemental Delta nodes can be configured as a cluster in order to support high availability and increased throughput. Input filters in a cluster should be configured to store content on shared storage. A single node in the cluster is denoted the Leader, and all input filters will be active on that node. If the Leader node fails, the second node will become the Leader and begin ingesting content. Endpoints can be accessed on either node in a cluster.

The web user interface and REST interface should only be updated on the current Leader node. The operator can configure a Virtual IP (VIP) address which will point to the active Leader node, and access control interfaces through the VIP.

Elemental recommends that clustered nodes are configured with static IP addresses on their management interfaces.

CONFIGURING A CLUSTER

The following steps are required to configure an Elemental Delta cluster:

- Install the same software version on two independent Elemental Delta systems.
- Select one node to be the initial Leader node. The other node will have all of its input filters, content, output filters, and output templates deleted during the clustering process.
- From the Nodes page on the Leader node, http://<node address>:8080/nodes, select “Add Node"
- Enter the hostname or IP address of the node you wish to add, and the shell password for the elemental user. This operation will overwrite the contents on the node to be added to the cluster. It will take several minutes for the node to be added to the cluster. Progress can be monitored by viewing the delta_cluster.log file.
- (Optional) Create a VIP to access the web user interface and REST interface.
- Once the cluster is created, configure the shared storage by clicking on the Settings icon for each node, and select Mount Points. Add a mount point to shared storage on each node, and ensure the Mount Folder is the same for each node.
- When configuring Input Filters, set the Storage Location to the Mount Folder to ensure content is written to shared storage accessible by both nodes in the cluster.

CLUSTER SETTINGS

Cluster configuration options can be set by clicking the Settings icon on the top right of the Nodes page on the cluster leader. See Cluster Settings for available options.

- REST Basics
  - HTTP Headers
  - API Versions
- Simple Examples
  - Input Filters
  - Output Templates
  - Output Filters
  - User Agent Presets
- Clean XML
- Input Filters
- Input Users
- Contents
- Output Templates
- Settings
- Alerts and Messages
- Clusters
- Authentication

WEB SERVICES REST INTERFACE

The Elemental Delta 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

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).

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>:8080/api/v1.0.2/contents will send a request to the /contents endpoint, and the server will interpret the data as compatible with Elemental API version 1.0.2. 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 /api/contents. 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. You can use cURL or a similar utility to interact with the REST interface. All the following examples use the curl utility.

INPUT FILTERS

To request a list of Input Filters from the server

```
curl -H "Accept: application/xml" http://<server_ip>:8080/api/input_filters
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<input_filter_list>
  <input_filter version="1.0.2.1" href="#1" product="Delta">
    <filter_type>webdav_input</filter_type>
    <label>default</label>
    <filter_settings>
      <id>1</id>
      <template_id/>
      <content_window_type>keep_seconds</content_window_type>
      <seconds_to_keep>7200</seconds_to_keep>
      <webdav_user_id>1</webdav_user_id>
      <relative_uri/>
      <storage_location>/data/server/content/</storage_location>
    </filter_settings>
  </input_filter>
  <input_filter version="0.5.0.1" href="#33" product="Delta">
    <filter_type>watch_folder_input</filter_type>
    <label>from xml</label>
    <filter_settings>
      <id>1</id>
      <template_id>1</template_id>
      <incoming>
      <id>3</id>
```

```
            <uri>/data/tmp/</uri>
            </incoming>
          </filter_settings>
        </input_filter>
      </input_filter_list>
```

To create a new WebDAV input filter:

```
curl -H "Accept: application/xml" -H "Content-type: application/xml" \
     -d @filename http://<server_ip>:8080/api/input_filters
```

where the file indicated by filename contains

```
    <input_filter>
      <label>WebDAV from xml</label>
      <filter_type>webdav_input</filter_type>
      <filter_settings>
        <template_id>1</template_id>
        <content_window_type>keep_seconds</content_window_type>
        <seconds_to_keep>120</seconds_to_keep>
        <webdav_user_id>1</webdav_user_id>
        <relative_uri/>
        <storage_location>/data/tmp/newest/</storage_location>
      </filter_settings>
    </input_filter>
```

returns

```
    <?xml version="1.0" encoding="UTF-8"?>
    <input_filter version="1.0.2.1" href="#35" product="Delta">
      <filter_type>webdav_input</filter_type>
      <label>WebDAV from xml</label>
      <filter_settings>
        <id>3</id>
        <template_id>1</template_id>
        <content_window_type>keep_seconds</content_window_type>
        <seconds_to_keep>120</seconds_to_keep>
        <webdav_user_id>1</webdav_user_id>
        <relative_uri/>
        <storage_location>/data/tmp/newest/</storage_location>
      </filter_settings>
    </input_filter>
```

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

OUTPUT TEMPLATES

To create a new output template

```
curl -H "Accept: application/xml" -H "Content-type: application/xml" \
   -d @filename http://<server_ip>:8080/api/output_templates
```

where the file indicated by filename contains

```
  <output_template>
    <label>Full Set</label>
    <filter>
      <name>filter_1</name>
      <endpoint>true</endpoint>
```

```
  <output_url>first_try_ar</output_url>
  <filter_type>ad_replace</filter_type>
  <filter_settings>
    <ad_sources>1</ad_sources>
  </filter_settings>
</filter>
<filter>
  <name>filter_2</name>
  <parent_filter>filter_1</parent_filter>
  <endpoint>true</endpoint>
  <output_url>first_try_bs</output_url>
  <filter_type>bitrate_selector</filter_type>
  <filter_settings>
    <selected_bitrate>
      <bitrate>290400</bitrate>
      <enabled>true</enabled>
    </selected_bitrate>
    <selected_bitrate>
      <bitrate>730400</bitrate>
      <enabled>true</enabled>
    </selected_bitrate>
    <selected_bitrate>
      <bitrate>2050400</bitrate>
      <enabled>true</enabled>
    </selected_bitrate>
  </filter_settings>
</filter>
<filter>
  <name>filter_3</name>
  <parent_filter>filter_1</parent_filter>
  <endpoint>true</endpoint>
  <output_url>first_try_blackout</output_url>
  <filter_type>blackout</filter_type>
  <filter_settings>
    <enabled>true</enabled>
    <content_ids>1</content_ids>
  </filter_settings>
</filter>
<filter>
  <name>filter_4</name>
  <parent_filter>filter_1</parent_filter>
  <endpoint>true</endpoint>
  <output_url>first_try_encrypt</output_url>
  <filter_type>hls_encryption</filter_type>
  <filter_settings>
    <encryption_type>AES-128</encryption_type>
    <key_rotation_count>3</key_rotation_count>
    <iv_follows_segment_number>true</iv_follows_segment_number>
    <constant_iv/>
    <key_format/>
    <key_format_versions/>
    <keyprovider_type>piksel</keyprovider_type>
    <keyprovider_settings>
      <content_ids>1</content_ids>
      <server>
        <uri>http://fakeserver.com</uri>
        <username>test</username>
        <password>test</password>
      </server>
    </keyprovider_settings>
  </filter_settings>
</filter>
<filter>
  <name>filter_5</name>
  <endpoint>true</endpoint>
```

```
      <output_url/>
      <filter_type>passthrough</filter_type>
    </filter>
    <filter>
      <name>filter_6</name>
      <parent_filter>filter_3</parent_filter>
      <endpoint>false</endpoint>
      <output_url/>
      <filter_type>time_delay</filter_type>
      <filter_settings>
        <delay_seconds>6</delay_seconds>
      </filter_settings>
    </filter>
    <filter>
      <name>filter_7</name>
      <parent_filter>filter_3</parent_filter>
      <endpoint>true</endpoint>
      <output_url/>
      <filter_type>user_agent</filter_type>
      <filter_settings>
        <user_agent_preset_ids>6,9</user_agent_preset_ids>
      </filter_settings>
    </filter>
    <filter>
      <name>filter_9</name>
      <parent_filter>filter_7</parent_filter>
      <endpoint>true</endpoint>
      <output_url/>
      <filter_type>vod_clip</filter_type>
      <filter_settings>
        <start_time>4</start_time>
        <end_time>7</end_time>
        <remove_ads/>
      </filter_settings>
    </filter>
  </output_template>
```

OUTPUT FILTERS

An HTTP PUT command on an existing Content object can be used to add new output filters. This example adds a Live to VOD filter and an MSS Packaging filter endpoint to a Live content.

```
curl -X PUT -H "Accept: application/xml" -H "Content-type: application/xml" \
   -d @filename http://<server_ip>:8080/api/contents/123
```

where the file indicated by filename contains

```
<content>
  <filter>
    <name>f1</name>
    <endpoint>false</endpoint>
    <filter_type>live_to_vod</filter_type>
    <filter_settings>
      <start_time>2014-10-30 12:30:00 -0700</start_time>
      <end_time>2014-10-30 13:00:00 -0700</end_time>
    </filter_settings>
  </filter>
  <filter>
    <name>f2</name>
    <parent_filter>f1</parent_filter>
    <endpoint>true</endpoint>
    <filter_type>mss_package</filter_type>
```

```
    <filter_settings>
      <fragment_duration>10</fragment_duration>
      <index_duration>60</index_duration>
    </filter_settings>
  </filter>
</content>
```

USER AGENT PRESETS

To request a list of User Agent Presets from the server

```
curl -H "Accept: application/xml" http://<server_ip>:8080/api/user_agent_presets
```

Response:

```
  <?xml version="1.0" encoding="UTF-8"?>
  <user_agent_preset_list>
    <user_agent_preset version="1.0.2.1" href="#1" product="Delta">
      <name>Android</name>
      <regex>Android</regex>
    </user_agent_preset>
    <user_agent_preset version="1.0.2.1" href="#2" product="Delta">
      <name>Chrome</name>
      <regex>Chrome</regex>
    </user_agent_preset>
    <user_agent_preset version="1.0.2.1" href="#3" product="Delta">
      <name>Firefox</name>
      <regex>Firefox</regex>
    </user_agent_preset>
    <user_agent_preset version="1.0.2.1" href="#4" product="Delta">
      <name>Internet Explorer</name>
      <regex>MSIE</regex>
    </user_agent_preset>
    <user_agent_preset version="1.0.2.1" href="#5" product="Delta">
      <name>iOS</name>
      <regex>(iPad)|(iPod)|(iPhone)</regex>
    </user_agent_preset>
    <user_agent_preset version="1.0.2.1" href="#6" product="Delta">
      <name>iPad</name>
      <regex>iPad</regex>
    </user_agent_preset>
    <user_agent_preset version="1.0.2.1" href="#7" product="Delta">
      <name>iPhone</name>
      <regex>iPhone</regex>
    </user_agent_preset>
    <user_agent_preset version="1.0.2.1" href="#8" product="Delta">
      <name>Macintosh</name>
      <regex>Macintosh</regex>
    </user_agent_preset>
    <user_agent_preset version="1.0.2.1" href="#9" product="Delta">
      <name>Windows</name>
      <regex>Windows</regex>
    </user_agent_preset>
  </user_agent_preset_list>
```

To create a new user agent preset:

```
 curl -XPOST -H "Accept: application/xml" -H "Content-type: application/xml" \
  http://<server_ip>:8080/api/user_agent_presets/ \
  -d "<user_agent_preset><name>Opera</name><regex>Opera</regex></user_agent_preset>"
```

returns

```
<?xml version="1.0" encoding="UTF-8"?>
<user_agent_preset version="1.0.2.1" href="#11" product="Delta">
  <name>Opera</name>
  <regex>Opera</regex>
</user_agent_preset>
```

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.

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 Output Templates, or to slightly modify Contents. Therefore, the Elemental Delta 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 Output Template 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>:8080/api/output_templates/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 Output Template, or the file may be edited to make any necessary adjustments. Clean XML for an Output Template can be downloaded from most web browsers by specifying the Output Template ID number as in this example:

```
http://<server_ip>:8080/api/output_templates/<Output Template ID number>.xml?clean=true
```

PAGINATION

Some GET requests for lists of objects return a paginated set of results. The parameters used to paginate the results can be adjusted by appending page and per_page parameters to the end of the request as follows:

```
/alerts?page=2&per_page=20
```

INPUT FILTERS

Input filters can be controlled with the following REST endpoints:

URL	METHOD	PARAMETERS	RETURNS	DESCRIPTION
/input_filters	GET		List of input filters	Retrieves current input filters in the system
/input_filters	POST	Input Filter Parameters	Input Filter Description	Creates a new input filter with the supplied settings.
/input_filters/<id>	GET		Input Filter Description	Retrieves details of a specific input filter in the system
/input_filters/<id>	PUT	Input Filter Parameters	Input Filter Description	Updates an existing input filter with the new settings.
/input_filters/<id>	DELETE			Deletes an input filter. Content created from this input filter will remain on the system.

INPUT USERS

Input users for WebDAV input filters can be controlled with the following REST endpoints:

URL	METHOD	PARAMETERS	RETURNS	DESCRIPTION
/input_users	GET		List of input users	Retrieves current input users in the system
/input_users	POST	Input User Parameters	Input User Description	Creates a new input user with the supplied settings.
/input_users/<id>	GET		Input User Description	Retrieves details of a specific input user in the system
/input_users/<id>	DELETE			Deletes input user <id>

REMOTE INPUT CONTENT

Content URLs on remote origin systems are specified for Remote Input Filters via the following API:

URL	METHOD	PARAMETERS	RETURNS	DESCRIPTION
/input_filters/<filter_id>/remote_input_contents	GET		List of content	Retrieves a list of the content managed by remote input filter <filter_id>
/input_filters/<filter_id>/remote_input_contents	POST	Remote Input Content Parameters		Creates a new Remote Content object managed by the remote input filter <filter_id>

CONTENTS

Content indexed by the system and their associated output filters can be controlled wit the following REST endpoints:

URL	METHOD	PARAMETERS	RETURNS	DESCRIPTION
/contents	GET		List of content	Retrieves a list of all content indexed by the system
/hls_contents	GET		List of HLS content	Retrieves a list of HLS content indexed by the system
/ts_contents	GET		List of TS content	Retrieves a list of MPEG transport stream content indexed by the system
/mss_contents	GET		List of MSS content	Retrieves a list of MPEG transport stream content indexed by the system
/contents/<id>	GET		Content Description	Retrieves details of a specific content item and its output filters and endpoints
/contents/<id>	PUT	Content Parameters	Content Description	Updates an existing content with the new settings.
/contents/<id>	DELETE			Deletes content. This will remove all filters, endpoints, stats, and will delete the content files from the filesystem.
/contents/<id>/add_filters	PUT	Content Parameters (with new filters to create)	Content Description, with only new filters that were created	Adds the new set of filters contained in the input content description. This API should be used to efficiently add new filters when a content has a large number of output filters.
/contents/<id>/filters	POST	Filter Parameters	Filter Description	Adds a single new filter to content <id> with the supplied parameters
/contents/<id>/filters/<id>	PUT	Filter Parameters	Filter Description	Updates the specified filter with the supplied parameters. Note this command can not be used to change the filter type. The user should delete the filter and create a new one to change type.
/contents/<id>/filters/<id>	DELETE			Deletes the specified output filter from the filter chain

OUTPUT TEMPLATES

Output templates can be controlled with the following REST endpoints:

URL	METHOD	PARAMETERS	RETURNS	DESCRIPTION
/output_templates	GET		List of output templates	Retrieves current output templates in the system
/output_templates	POST	Output Template Parameters	Output Template Description	Creates a new output template with the supplied settings.
/output_templates/<id>	GET		Output Template Description	Retrieves details of a specific output template in the system
/output_templates/<id>	DELETE			Deletes output template <id>

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	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/network	GET	Network Settings	Retrieves information about the current network settings. Other Elemental Delta 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.

ALERTS AND MESSAGES

The alerts API provides information about current alert conditions on the Elemental Delta unit. Messages provide an audit list of events on Delta

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=Audit	List of messages	Messages can be Content, Error, Warning, or Audit messages. They have a code and a text message.

CLUSTERS

Cluster and VIP settings can be controlled with the following REST endpoints:

URL	METHOD	PARAMETERS	RETURNS	DESCRIPTION
/clusters/1	GET			Current cluster settings. Note Cluster ID will always be 1
/clusters/1	PUT	Cluster Parameters		Updates the cluster settings. Note Cluster ID will always be 1
/clusters/1/vips	GET		List of active VIPs	List of active Virtual IP (VIP) addresses

URL	METHOD	PARAMETERS	RETURNS	DESCRIPTION
/clusters/1/vips	POST	VIP Parameters	VIP Description	Creates a new VIP with the supplied settings
/clusters/1/vips/<id>	GET		VIP Description	Retrieves details of a VIP
/clusters/1/vips/<id>	DELETE			Deletes VIP <id>

AUTHENTICATION AND REST

When authentication is enabled on the Elemental Delta 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 http://deltaserver:8080/contents/1?clean=true by the user 'admin' with the api_key '1acpJN7oEDn3BDDYhQ' that expires on June 1, 2014 UTC. In this case the url parameter is '/contents/1' and the X-Auth-Expires value is '1401580800'. Thus the value of X-Auth-Key should be computed as follows:

```
md5('1acpJN7oEDn3BDDYhQ' + md5('/contents/1'+'admin'+'1acpJN7oEDn3BDDYhQ'+'1401580800'))
          => md5('1acpJN7oEDn3BDDYhQ' + md5('/contents/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 '/contents/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' http://deltaserver:8080/contents/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 \
          -H 'Accept: application/xml' -H 'Content-Type: application/xml' \
          -d @filename http://deltaserver:8080/input_filters
```

- Location
- Input Filter
  - UDP Input Filter Settings
  - UDP Input
  - Watch Folder Settings
  - WebDAV Settings
  - MSS Settings
  - Remote Input Filter
  - Input User
- Content
- Remote Input Content
- Filter
- Stream Set
  - Video Track
  - Audio Track
  - Subtitle Track
  - Track Selector
- Passthrough Filter
- Ad Removal Settings
- Ad Replace Settings
- Akamai G2O Authentication
  - Nonce Key Pair
  - Whitelist Entry
- Bitrate Selector Settings
  - Selected Bitrate
- Blackout Settings
- Cache Settings
- Cisco URL Signing Settings
- Common Encryption
- DASH-ISO Package Filter Settings
- File Copy Filter Settings
- Flash Access Filter
- HDS Package Filter Settings
- HLS Encryption Settings
  - Verimatrix Settings
  - Secure Media Settings
  - Irdeto Settings
  - Conax Settings
  - Generic Keyprovider Settings
  - Piksel Settings
  - VOSP Settings

PARAMETERS

The following tables outline parameters that can be set for objects in Elemental Delta. These can be set using REST or the appropriate pages in the web interface. The Name column contains the appropriate XML tag for each parameter, and names in bold are required fields. If there is a specific range of valid values for a parameter, it will be displayed in the Range column.

LOCATION

Location is an object used in several parameters to define a filesystem or network resource that may optionally include user credentials.

For example, a watch folder's incoming parameter is specified as follows:

<incoming> <uri>/data/server/outgoing/</uri> </incoming>

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
uri	string			Uniform Resource Identifier – This should be a path to a file accessible to the Elemental Delta system, either on the local filesystem or through a mount point, or a URI depending on the output type. For example, a key server should have a uri simliar to: “http://keyserver/”.
username	string			Username if credentials are required to access file or publishing point.
password	string			Password if credentials are required to access file or publishing point.

INPUT FILTER

Input Filters define ingest points for content received by the system. They can control storage locations for content, retention windows, and content types.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
label	string			Name to use for identification purposes. For UDP Input Filters, this label will be copied to the Content object associated with this input filter.
filter_type	string	webdav_input, watch_folder_input, udp_input, mess_input, remote_input		The type of input filter that the settings are for.
filter_settings		WebDAV Settings, Watch Folder Settings, UDP Input Filter Settings, MSS Settings, Remote Input Filter		Settings for each of the input filter types. The accepted settings are defined in the settings that match the type.

UDP INPUT FILTER SETTINGS

A UDP Input Filter allows adaptive bitrate (ABR) content to be delivered to the Elemental Delta system as single-program MPEG Transport Streams over unicast or multicast UDP. The streams may have RTP headers. Content is stored on the filesystem as specified by the user. Each UDP Input Filter generates a corresponding TS Content entry in the Content page. One or more just-in-time packaging output filters should be applied to this content for delivery to end users.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
template_id	integer	Valid Output Template ID		Output Template to apply when content is sent to the UDP filter.
storage_location	string			Folder to store the incoming content to.
segmentation_marker		idr, ebp	idr	Type of segmentation marker in input stream. IDR segments at each closed GOP boundary. EBP segments at Encoder Boundary Points as specified by OpenCable OC-SP-EBP-I01-130118
content_window_type	string			Rule for determining how much content to keep on disk. If keep_seconds then the seconds_to_keep needs to be specified, and if keep_all then no content will be deleted automatically.
seconds_to_keep	integer			Required if content_window_type is set to keep_seconds. This value determines how many seconds of content to keep on disk.
enable_fec_rx	boolean		false	Enables SMPTE 2022-1 and SMPTE 2022-2 (ProMPEG) FEC reception on input stream. If FEC data is not received, input will function, but an error will be logged. Only compatible with RTP inputs.
buffer_time	integer		30	On non-leader node in a cluster, buffer this many seconds of incoming content. When the node becomes leader in a failover, it will insert this content to compensate for content lost in the failure detection. This value should be higher than the Drop Node After value for the cluster. Larger values will consume more memory on the system.
udp_inputs	UDP Input			List of URIs to receive content on.

UDP INPUT

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
uri	string			URI of UDP or RTP input to ingest. Should contain an IP address and port (Example: udp://239.255.1.10:5001). Note if the firewall is enabled, the user should add this port to the list of open incoming ports in http://<server ip>/settings/firewall. Streams with RTP headers should be specified as rtp://<ip>:<port>.
igmp_source	string	Valid IP address		Source address for Source Specific Multicast streams.
bitrate	integer	>0	nil	Bitrate for this input stream. Packaging filters will use this value for adaptive bitrate manifest creation. If left blank, the system will autodetect a value.

WATCH FOLDER SETTINGS

A Watch Folder Input Filter defines a directory accessible to Elemental Delta. The directory is monitored by the system for new input media files. When a new set of content is placed in a folder associated with a Watch Folder Input Filter, Elemental Delta will ingest the content and apply the associated Output Template. The directory can either be on the Elemental Delta system's disk drives or an external storage system mounted in the Settings → Mount Points page. Watch Folder Input Filters can be used for VOD content.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
template_id	integer	Valid Output Template ID		Output Template to apply when content appears in the watch folder.
incoming	Location			Folder to watch for new content.
search_subfolders	boolean			Search subfolders when true, otherwise don't.

WEBDAV SETTINGS

WebDAV Input Filters allow content to be pushed to the Elemental Delta system over HTTP WebDAV (http://en.wikipedia.org/wiki/WebDAV). To define a WebDAV input filter, the operator configures a relative path, a set of user credentials, and a disk location to store incoming content. The operator should provide an upstream encoder or packager with the Absolute Path and the Username and Password of the selected input user. The operator can also apply an Output Template to new content.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
template_id	integer	Valid Output Template ID		Output Template to apply when content is sent to WebDAV.
relative_uri	string			Appends a path to the WebDAV URI. (eg, http://<web server>/in/<relative_uri>).
input_user_id	integer	Valid Input User ID		Input User to use for permissions to this uri.
storage_location	string			Folder to store the incoming content to.
vod_content	boolean			Ingests content as VOD content. This allows Delta to ingest VOD content as it is being created by the upstream packager.
content_window_type	string	packager_controlled, keep_seconds, keep_all		Rule for determining how much content to keep on disk. If packager_controlled then DELETE requests are honored in WebDAV, if keep_seconds then the seconds_to_keep needs to be specified, and if keep_all then no content will be deleted automatically.
seconds_to_keep	integer			Required if content_window_type is set to keep_seconds. This value determines how many seconds of content to keep on disk.

MSS SETTINGS

MSS Input Filters allow Microsoft Smooth Streaming content to be pushed to the Elemental Delta system over HTTP. To define an MSS input filter, the operator configures a relative path and a disk location to store incoming content. The operator should provide an upstream encoder or packager with the Absolute Path to deliver content. The operator can also apply an Output Template to new content.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
template_id	integer	Valid Output Template ID		Output Template to apply when content is sent to the MSS Input Filter.
relative_uri	string			Appends a path to the WebDAV URI. (eg, http://<web server>/in_mss/<relative_uri>).
storage_location	string			Folder to store the incoming content to.
content_window_type	string	keep_seconds, keep_all		Rule for determining how much content to keep on disk. If keep_seconds then the seconds_to_keep needs to be specified, and if keep_all then no content will be deleted automatically.
seconds_to_keep	integer			Required if content_window_type is set to keep_seconds. This value determines how many seconds of content to keep on disk.

REMOTE INPUT FILTER

Remote Input Filters allow the processing and delivery of content that is originated by another web server. Elemental Delta will fetch content from the remote origin location and process it using the output filter chain. Segments that are retrieved from the remote origin are cahced in memory and disk caches configured in the Remote Input Filter for subsequent requests.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
remote_uri	string			Base URI of remote origin server to fetch content from. Individual content paths are appended to this value.
disk_cache_size	integer			Disk cache size in Megabytes for this input filter
memory_cache_size	integer			Memory cache size in Megabytes for this input filter
template_id	integer	Valid Output Template ID		Output Template to apply when new remote content is created
storage_location	string			Storage location for disk cache
bind_filters_to_output_template	boolean		false	Re-creates output filters when output template changes. This will delete all output filters on each content associated from this input filter when the output template is updated.
vod_content	boolean		false	Treat remote content as VOD content. This allows Delta to use VOD content as it is being created.

INPUT USER

Input Users are the credential sets (name and password) to authorize ingestion for a WebDAV Input Filter. They have no bearing on who accesses the Delta system.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
username	string			Username for input user.
password	string			Password for input user.

CONTENT

Content managed by the Elemental Delta system

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
output_template_id	integer	Valid Output Template ID		If defined, will assign the filters from the Output Template to the Content
name	string			A user-defined identifier for the Content.
path	string			If the content is coming from a source other than the input filters, then the path to the manifest is required. If a path is already defined then the path cannot be changed.
encrypted	boolean			True if the content is encrypted.
filter	Filter			One or more filters that can be chained together to form the endpoints for this content.

REMOTE INPUT CONTENT

Content associated with a Remote Input Filter

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
uri	string	string		Path to remote content. The path is appended to the remote_uri of the associated Remote Input Filter.
name	string			A user-defined identifier for the Content.

FILTER

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
name	string			Name that descendant filters can reference to attach to this filter. This value is only used for filter set creation and is not retained once the filter is created; refer to filter ID once the filter has been created.
description	string			Any descriptive information you want to add. This information will show up as a tooltip on hover in the UI.
parent_filter	string			Name of parent filter if this filter is a descendant of another filter. If empty, this is a top-level filter attached directly to the content. Use parent_filter if the parent filter has not yet been created.
parent_id	integer	Existing output filter ID		ID of parent filter if this filter is a descendant of another filter. If empty, this is a top-level filter attached directly to the content. Use parent_id if the parent filter has already been created.
endpoint	boolean			Allows the endpoint to be publicly available and has a default access of http://<web server>/out/i/<filter_id>.m3u8 (http://<web server>/out/i/<filter_id>/<file_name>.m3u8 for passthrough filters).
output_url	string			If the endpoint is enabled and the output_url is defined then there will be a second, user defined, endpoint available at the url http://<web server>/out/u/<output_url>.m3u8 (http://<web server>/out/u/<filter_id>/<output_url>.m3u8 for passthrough filters).

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
filter_type	string	ad_removal, ad_replace, akamai_g2o_authentication, blackout, bitrate_selector, cisco_url_signing, flash_access, hds_package, hls_encryption, hls_package, live_to_vod, mss_package, passthrough, playready, time_delay, user_agent, vod_clip		Defines the type of filter that the settings are defined for.
filter_settings	Filter Settings	Ad Removal Settings, Ad Replace Settings, Akamai G2O Authentication, Blackout Settings, Bitrate Selector Settings, Cache Settings, Cisco URL Signing Settings, Common Encryption, DASH-ISO Package Filter Settings, File Copy Filter Settings, Flash Access Filter, HDS Package Filter Settings, HLS Encryption Settings, HLS Package Filter Settings, MP4 Package Filter Settings, MPEG-TS Package Filter Settings, MSS Package Filter Settings, Playready Settings, Time Delay Settings, User Agent Settings, VOD Clip Settings, Live to VOD Settings		Filter Specific Settings. The accepted settings are defined in the settings that match the type (no settings required for passthrough).
use_default_stream_sets	boolean		true	When enabled, input tracks will be assigned to appropriate output stream sets for this packaging filter. Stream sets will be updated when input tracks change. When disabled, stream sets should be created by the operator, and must be updated manually if input tracks change.
stream_set	Stream Set			One or more Stream Set definitions for packaging filters.

STREAM SET

Stream Sets are attached to packaging filters to select the input video and audio streams to use for the output. When used in an output template, the user can select by one of the following parameters: bitrate range, incoming track index, PID (for UDP and HLS content types), or language for audio tracks. If new content being applied to this stream set does not match the requested streams, an alert will be generated.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
stream_index	integer			The order which this Stream Set should appear in the output manifest.
video_track	Video Track	A list of video tracks to include in this Stream Set		Specification for video track in this stream_set. MSS Package filters can have multiple video_track items in a Stream Set; all other package filters should have only one video_track per stream_set.
audio_track	Audio Track	A list of audio tracks to include in this Stream Set		Specification for audio track in this stream_set.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
subtitle_track	Subtitle Track	A list of subtitle tracks to include in this Stream Set		Specification for subtitle track in this stream_set.
rendition_group_name	string			Allows this stream set to be configured as an alternate rendition group for HLS Package filters. Valid only for stream sets with subtitle tracks or multiple audio tracks.
iframes_only	boolean		false	Produces an EXT-X-I-FRAMES-ONLY stream. Valid only for stream sets in HLS package filters with a single video track.

VIDEO TRACK

Video Tracks can be specified using a track_id when used in a filter attached to incoming content and the user knows the incoming track's id. When specified in a filter attached to an Output Template, a Track Selector is specified.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
track_index	integer			The order which this track should appear in the Stream Set.
track_id	integer	Valid track ID in content		When attached to a filter on existing content, video track can be defined by incoming track ID.
track_selector	Track Selector			When applied to a filter in an output template or attached to content whose input has not yet begun, a track_selector must be used.
rendition_group_value				Name of this video track

AUDIO TRACK

Audio Tracks can be specified using a track_id when used in a filter attached to incoming content and the user knows the incoming track's id. When specified in a filter attached to an Output Template, a Track Selector is specified.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
track_index	integer			The order which this track should appear in the Stream Set.
track_id	integer	Valid track ID in content		When attached to a filter on existing content, audio track can be defined by incoming track ID.
track_selector	Track Selector			When applied to a filter in an output template or attached to content whose input has not yet begun, a track_selector must be used.
rendition_group_value				Name of this audio track when used in an alternate rendition group. Required if stream_set has rendition_group_name set.

SUBTITLE TRACK

Subtitle Tracks can be specified using a track_id when used in a filter attached to incoming content and the user knows the incoming track's id. When specified in a filter attached to an Output Template, a Track Selector is specified.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
track_index	integer			The order which this track should appear in the Stream Set.
track_id	integer	Valid track ID in content		When attached to a filter on existing content, subtitle track can be defined by incoming track ID.
track_selector	Track Selector			When applied to a filter in an output template or attached to content whose input has not yet begun, a track_selector must be used.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
rendition_group_value				Name of this subtitle track when used in an alternate rendition group. Required if stream_set has rendition_group_name set.

TRACK SELECTOR

Specifies a set of criteria to match tracks on incoming content with appropriate tracks in a Stream Set. A track_selector should specify only one of the selection criteria available.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
input_stream_index	integer			Selects track by index in the incoming content relative to other streams.
pid	integer	0 - 8192		PID of incoming stream. This selection criteria is only valid for filters attached to HLS and TS content.
codec	string	AACL, AACH, DFXP, H264, H265, TTML, WebVTT		Codec of incoming track. Note there must only be one stream of the specified code in the incoming content to select on codec.
language	string	ISO 639-2 three-digit code		Language of incoming track.
minimum_bitrate	integer			Select a track with a bitrate greater than this bitrate. Must be used in conjunction with maximum_bitrate to supply a range, as incoming bitrates may not be detected precisely.
maximum_bitrate	integer			Select a track with a bitrate less than this bitrate. Must be used in conjunction with minimum_bitrate to supply a range, as incoming bitrates may not be detected precisely.

PASSTHROUGH FILTER

A passthrough filter will serve the content to the end user as it was formatted by the upstream packager. Passthrough is supported for HLS and MP4 content. Note that if the input filter's content window is smaller than the packager's, playback may be disrupted.

AD REMOVAL SETTINGS

Ad Removal filters can be attached to VOD content to remove ads. Input streams must be decorated with #EXT-CUE-OUT and #EXT-CUE-IN ad markers.

AD REPLACE SETTINGS

Ad replace filters replaces the main content stream with the specified advertising content during ad avails. Input streams must be decorated with #EXT-CUE-OUT and #EXT-CUE-IN ad markers.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
ad_sources	string	List of Valid Content IDs		A comma separated list of Content IDs to use for ad replacement. Advertising content should be VOD content formatted to match ad avail timing – for instance if ad avails are 60 seconds, ad content should be in 15, 30, or 60 second intervals. The system will move through the list of content as ad avails appear in the stream, and restart at the beginning of the list once all ads have been inserted into the stream.

AKAMAI G2O AUTHENTICATION

This filter protects requests using Akamai Signature Header Authentication. Elemental Delta will only serve content from this filter when requested from an Akamai edge server with the correct authentication headers.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
clear_data_header_name	string		X-Akamai-G2O-Auth-Data	HTTP header containing clear G2O settings.
encrypted_data_header_name	string		X-Akamai-G2O-Auth-Sign	HTTP header containing G2O signature.
request_validity_window	integer	> 0	30	Amount of time in seconds to consider a G2O request valid. Allows the user to correct for time differences between Delta machine and Akamai edge servers.
nonce_key_pairs	Nonce Key Pair			A list of nonce_key_pair objects to consider for signature validation. These must be set to equal to the nonce and key pair in the Akamai edge configuration. Additionally, if the edge configuration changes, a new nonce_key_pair should be added and the previous one should be retained until the transition to the new key is complete.
akamai_g2o_whitelist_entries	Whitelist Entry			A list of whitelist_entry objects to bypass for signature validation.

NONCE KEY PAIR

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
nonce	string			Nonce identifies a key used for generating the signature.
key	string			Encryption key used for generating the signature.

WHITELIST ENTRY

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
ip_address	string			Client IP Address to bypass Akamai G2O authentication. This can be a single IP address or a range in CIDR notation.

BITRATE SELECTOR SETTINGS

Bitrate filters allow the operator to serve only certain bitrates from a set of adaptive bitrate streams. When attached to content, the operator can enable and disable individual bitrates by toggling the selected bitrates. When used in an Output Template, the operator can specify minimum and maximum bitrates to serve. Five megabits can be entered as 5000000 or 5m. Five hundred kilobits can be entered as 500000 or 0.5m. When the template is applied to new content, only bitrates in this range will be selected.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
selected_bitrates	Selected Bitrate			Set of bitrates that will be enabled/disabled for the content that the filter is applied to.

SELECTED BITRATE

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
bitrate	integer			Bitrate to be enabled or disabled. If a bitrate is not defined it is disabled by default.
enabled	boolean			If enabled then the bitrate will be available if the Bitrate Selector Filter is an endpoint.

BLACKOUT SETTINGS

Blackout filters are used to select some content to be displayed when the program needs to be blacked out. The blackout functionality can be enabled/disabled (i.e. the replacement content or the original is displayed) upon an API call, or through the GUI.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
content_ids	string	List of Valid Content IDs		A comma separated list of Content IDs to display when the blackout filter is enabled.
enabled	boolean		false	Enable blackout. When set, blackout content will be served instead of the program content.

CACHE SETTINGS

Allows override of default max-age headers for all endpoints in this filter chain

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
max_age_vod_variant_manifest	integer		21600	Time (in seconds) a downstream server should cache VOD variant manifest. This applies to top-level HLS and HDS manifests for VOD content or after a completed Live to VOD filter.
max_age_vod_bitrate_manifest	integer		21600	Time (in seconds) a downstream server should cache VOD bitrate manifest. This applies to top-level MSS and DASH manifests and HLS and HDS bitrate manifests for VOD content or after a completed Live to VOD filter.
max_age_live_variant_manifest	integer		2	Time (in seconds) a downstream server should cache live variant manifest. This applies to top-level HLS and HDS manifests for live content.
max_age_live_bitrate_manifest	integer		2	Time (in seconds) a downstream server should cache live bitrate manifest. This applies to top-level MSS and DASH manifests and HLS and HDS bitrate manifests for live content.
max_age_content_files	integer		21600	Time (in seconds) a downstream server should cache content files

CISCO URL SIGNING SETTINGS

Adds URL signatures compatible with Cisco Internet Streamer CDS to bitrate playlist URLs for HLS outputs. This filter only adds signatures, it does not perform validation.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
url_prefix	string			Prefix of URL to be signed. This should correspond to the public URL the content will be accessed from
client_ip	string	Valid IP Address	1.2.3.4	Client IP address to generate signature for. Field is still required if client IP validation is disabled, and can be any valid IP address.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
expiration_window	integer			Time (in seconds) from now that the URL signature will be valid for
key_owner_id	integer	1 to 32		Key Owner for key lookup
key_number_id	integer	1 to 16		Key Number for key lookup
key	string			Key to generate signature
exclude_domain	boolean		false	Exclude domain from URL signature
version	integer	0, 1, or 2		Version of signature algorithm

COMMON ENCRYPTION

Common Encryption filters are used to apply CENC DRM/encryption algorithm to MPEG DASH-ISO output content.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
keyprovider_type	string	widevine	widevine	Keyprovider for CENC
license_url	string			URL for keyprovider server
content_id	string			Unique content identifier for this output filter
provider_id	string			Signer for Widevine keyprovider requests
provider_key	hexadecimal string			AES key for signing Widevine keyprovider requests. This is a 256-bit hex value represented by a 64-character string.
provider_iv	hexadecimal string			AES Initialization Vector (IV) for signing Widevine keyprovider requests. This is a 128-bit hex value represented by a 32-character string.
key_rotation_count	integer		0	Number of segments before requesting a new key from keyprovider. Set to 0 to use the same key for duration of the content

DASH-ISO PACKAGE FILTER SETTINGS

Packages content for the DASH-ISO ABR streaming protocol. Can be applied to any content type. Endpoints can be accessed at http://<server ip>/out/i/<endpoint id>.mpd.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
fragment_duration	integer	greater than zero		Time duration (in seconds) of each generated fragment. Actual fragments will be rounded to the nearest multiple of source fragment duration.
index_duration	integer	greater than zero		Length of time (in seconds) to emit in manifest for live content. Parameter is ignored for VOD content.
hbbtv	string	empty or 1.5		Set to 1.5 to enable HbbTV compliant output.
min_update_period	integer	greater than zero		Minimum time (in seconds) between potential changes to the MPD.
min_buffer_time	integer	greater than zero		Specifies the minimum length of time (in seconds) that the client (player) will buffer media before starting the presentation.

FILE COPY FILTER SETTINGS

When placed on file-oriented content (such as MP4) or after a file-oriented packaging filter (such as MP4 Package or MPEG-TS Package), the File Copy filter will copy the file(s) to a directory of your choosing. If a directory name (a name ending in a '/') is specified (in the path field) then the filter id will be used as the base name for files. If, on the other hand, a name is given after the '/' then that will become the base name for files. Subdirectories will be created if needed. See the MPEG-TS and MP4 Package filters for naming conventions for ABR content.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
path	string	directory or file path		Specify an existing directory (ending in '/') and the filter will output files named with the filter id as the base name into that directory. Specify a name after the '/' and the filter will use that as the base name for files created in that directory. Filter will create directories if they don't already exist.

FLASH ACCESS FILTER

Applies Adobe Flash Access encryption upon client request. Can apply Flash Access or pHDS encryption to HDS packaging filters, or Flash Access or pHLS to HLS content or HLS packaging filters.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
use_phds	boolean		true	Enables Protected HDS or Protected HLS (when applied to HLS content or an HLS package filter). When true, no other parameters are required. When false, applies full Flash Access encryption with user's Flash Access server.
encrypt_video	boolean		true	Encrypt video data.
encrypt_audio	boolean		true	Encrypt audio data.
encrypt_data	boolean		true	Encrypt stream metadata.
generate_cek	boolean		true	When checked, the Encryption Key and Content ID are combined to generate a unique Content Encryption Key (CEK). When unchecked, the Encryption Key is used directly as the CEK. When checked, the Encryption Key can be a file of arbitrary length. When unchecked, the Encryption Key file must be 16 bytes (128 bits) long.
video_encrypt_level	string	low, medium, high	low	Indicates the degree of partial encryption to apply. Low implies the lowest amount of partial encryption should be applied. A subset of the samples (like video keyframes) are encrypted. Medium implies a medium amount of partial encryption should be applied. High implies full encryption.
content_id	Location			When a common key is in use the content_id is used along with the encryption_key to generate a content encryption key.
license_server	Location			The URL of the license server used for protecting content.
content_encryption_key	Location			The cryptographic key used to encrypt the content.
license_server_certificate	Location			A unique certificate file obtained from Adobe which identifies the license server, in DER format.
license_server_credential	Location			Credentials for the Adobe HDS license server.
license_server_credential_password				Password for credential file
packager_credential	Location			Credentials for the Adobe HDS packager.
packager_credential_password				Password for credential file
transport_certificate	Location			The transport certificate, in DER format.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
policy_file	Location			A file which contains the rules and restrictions that determine how, when, and where protected content can be viewed by consumers.
swf_identifiers	Location			Specifies a file of hashes of SWF players that are approved players for this content. Use the Adobe Media Server whitelist tool to generate these files.
key_server_certificate	Location			Certificate required to support an embedded (non-chained) license with Remote Key Delivery.
recipient_certificate	Location			A certificate which uniquely identifies the recipient machine and client instance.

HDS PACKAGE FILTER SETTINGS

Packages content for Adobe HTTP Dynamic Streaming. Can be applied to any content type. HDS package filter endpoints can be accessed at http://<server ip>/out/i/<endpoint id>.f4m

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
fragment_duration	integer			Duration in seconds of f4f fragments to create. Actual fragments will be rounded to a multiple of the source content's fragment length.
index_duration	integer			Length in seconds to emit for a live manifest.
external_bootstrap	boolean			Emit bootstrap as a separate file. If false, bootstrap will be embedded in manifest file. Some HDS players require external bootstrap files.
avail_trigger	string	all, placement_opportunity		all will generate cue tags based on all SCTE-35 ad types. placement_opportunity will generate them only for placement opportunities.
ad_markers	string	none, dpi_simple, dpi_scte35		Select the Ad Marker format to appear in HDS outputs. Specifying DPI Simple (dpi_simple) or DPI SCTE-35 (dpi_scte35) will add a cueInfo tag to the HDS manifest containing cue information formatted in the corresponding Adobe DPI profile, and will also cause OnCuePoint and OnCuePointContinuation AMF messages to appear in the HDS data stream.
broadcast_time	boolean			When true, ID3Timed Metadata messages are generated every 5 seconds with the ingest time of the content. Irrespective of this parameter, if any ID3 Timed Metadata is found in HLS input is translated to an OnCuePointBroadcastTime in HDS output.
ignore_web_delivery_allowed	boolean			When true, SCTE-35 segmentation descriptors with web_delivery_allowed_flag set to 0 will no longer trigger blackouts or avails.
ignore_no_regional_blackout	boolean			When true, SCTE-35 segmentation descriptors with no_regional_blackout set to 0 will no longer trigger blackouts or avails.
absolute_timestamps	boolean			Each HDS timestamp will specify the time the corresponding segment was ingested by Elemental Delta. Times are expressed in milliseconds relative to a base time. When absolute_timestamps is true, the base time is the epoch (00:00:00 UTC, Thursday, 1 January 1970). Otherwise the base time is the start of the stream.

HLS ENCRYPTION SETTINGS

HLS Encryption filters are used to apply a DRM/encryption algorithm to the content when requested by an end user. Elemental Delta is integrated with several key providers.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
key_rotation_count	integer			The AES-128 encryption key will rotate after this many segments. Set to 0 to use the same key for the duration of the content. This parameter will be ignored when the key provider is Conax or Irdeto.
iv_follows_segment_number	boolean		true	The IV (Initialization Vector) is a 128-bit number used in conjunction with the key for encrypting blocks. If this setting is enabled, it will cause the IV to change every segment (to match the segment number). If this is set to false, you must enter a constant_iv value.
constant_iv	hexadecimal string			This is a 128-bit, 16-byte hex value represented by a 32-character string. If iv_follows_segment_number is set to false then this parameter is required and is used as the IV for encryption.
key_format	string	identity, com.example.foo		If left empty 'identity' is implied. A reverse DNS string can also be given.
key_format_versions	string	1, 1/2/3, 1/3		Either a single positive integer version value or a slash delimited list of version values (1/2/3).
keyprovider_type	string	verimatrix, secure_media, irdeto, conax, generic_keyprovider, piksel, vosp, self_generated		Keyprovider type.
keyprovider_settings	Key Provider Settings	Verimatrix Settings, Secure Media Settings, Irdeto Settings, Conax Settings, Generic Keyprovider Settings, Piksel Settings, VOSP Settings, Self-Generated Settings		Key Provider-specific settings
custom_attributes				Custom attributes.

VERIMATRIX SETTINGS

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
resourceid	string			Verimatrix Resource ID.
resource_type	string	dtv or vod	vod	Verimatrix Resource type. Use dtv for live content, and vod for VOD content or after Live to VOD filters.
verimatrix_server	Location			The Verimatrix server that will provide the keys.
reuse_last_key	boolean		true	If checked, the stream will be encrypted using the last key obtained from the Verimatrix Server in the event that server becomes unreachable.

SECURE MEDIA SETTINGS

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
resourceid	integer	0 – 4294967295		Secure Media Resource ID.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
secure_media_server	Location			The Secure Media server that will provide the keys.
reuse_last_key	boolean		true	If checked, the stream will be encrypted using the last key obtained from the SecureMedia server in the event that the server becomes unreachable.

IRDETO SETTINGS

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
service_url	Location	URL with login credentials		Specifies the Location of the Irdeto server. Both a URL and login credentials are required.
static_key	boolean		false	When true, the operator provides key_id, key_seed, la_url, and domain_service_id. When false, those values are fetched from the Irdeto server
account_id	string			Used to identify the account on the Irdeto Control server.
content_id	string			Used to identify the content in Irdeto Control so that the content key can be associated.
content_key	string	generate_new_key or use_last_key	generate_new_key	Determines if a new key should be generated at the start of encoding or if the encoding session should use the last key.
use_https	boolean		false	Specifies whether requests to the License Acquisition URL should use HTTPS or basic HTTP.
sub_content_type	string	default, SSPlayReady, HLSPlayReady or other customer supported values.		Specifies the sub content type to be associated with the output group.
key_id	string	GUID		Specifies a key ID when static_key is enabled. Must be a valid GUID.
key_seed	string			Contains a base64-encoded key seed. Used only when static_key is enabled
domain_service_id	string	GUID		Service ID. Must be a valid GUID. Used only when static_key is enabled
la_url	string			License Acquisition URL. Used only when static_key is enabled

CONAX SETTINGS

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
server	Location	URL with login credentials		Specifies the Location of the Conax Server. Both a URL and login credentials are required.
content_id	string			Used to identify the content on the Conax Server.

GENERIC KEYPROVIDER SETTINGS

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
server	Location			Specifies the Location of the Generic Keyprovider server. A valid URI is required. Optional username and password are used if the keyprovider requires authentication.
resourceid	string			Used by the Generic Keyprovider to identify the content.
reuse_last_key	boolean		true	If checked, the stream will be encrypted using the last key obtained from the key provider in the event that the key provider becomes unreachable.

PIKSEL SETTINGS

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
server	Location			Specifies the Location of the Piksel Server. Optional username and password are used if the keyprovider requires authentication.
content_id	string			Used to identify the content on the Piksel Server.

VOSP SETTINGS

For testing purposes, you may provide a content key directly via the content_key_hex or content_key_base64. Do not provide both a content_key_hex and a content_key_base64—just provide one or the other or leave them both blank. When they are both blank, Elemental Delta will acquire a content key from the service designated by the server_url parameter.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
server_url	URL			The URL to query for the VOSP content key.
service_id	string			The VOSP ServiceID.
license_url	URL			The URL for the license acquisition Web service.
ui_license_url	URL			The URL for a non-silent license acquisition Web page.
key_id	GUID			Specifies a key ID to use for Playready DRM, must be a valid GUID.
content_key_base64	string			A base64-encoded content key. If present, server_url is not used to retrieve key.
content_key_hex	string			Contains a hexadecimal-encoded content key. If present, server_url is not used to retrieve key.
custom_attributes	string			The content author can add arbitrary custom attributes inside this element. Microsoft code does not act on any data contained inside this element.

SELF-GENERATED SETTINGS

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
common_key	boolean		false	When enabled, generates the same key for each output within this output group.
key_prefix	string			A partial URI prefix that will be prepended to the key filenames in the output manifest. The prefix should point to the final publishing destination for the keys.

MP4 PACKAGE FILTER SETTINGS

Packages content in the MP4 (file) format. Can be applied to any content type. Works with VOD content or downstream of a Live To VOD filter on live content. If multiple bitrates are selected you'll get multiple .mp4 files. Names will have stream index appended in that case e.g. moonwalk_1.mp4, moonwalk_2.mp4. Use File Copy filter to save the files to the filesystem. The files are avialable via HTTP progressive download on endpoints like http://<server ip>/out/i/<endpoint id>.mp4 SMIL manifest files may be accessed at http://<server ip>/out/i/<endpoint id>.smil

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
major_brand	string			If present, this value overrides the Major Brand field in the output file.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
include_cslg	boolean			When true, file composition times will start at zero, composition times in the ‘ctts’ (composition time to sample) box for B-frames will be negative, and a ‘cslg’ (composition shift least greatest) box will be included per 14496-1 amendment 1. This improves compatibility with Apple players and tools.

MPEG-TS PACKAGE FILTER SETTINGS

Packages content in the MPEG-TS (file) format. Can be applied to any content type. Works with VOD content or downstream of a Live To VOD filter on live content. If multiple bitrates are selected you'll get multiple .ts files. Names will have stream index appended in that case e.g. marsmission_1.ts, marsmission_2.ts. Endpoints can be accessed via HTTP progressive download at http://<server ip>/out/i/<endpoint id>.ts SMIL manifest files may be accessed at http://<server ip>/out/i/<endpoint id>.smil

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
bitrate	integer		0	The output bitrate of the transport stream in bits per second. Setting to 0 lets the muxer automatically determine the appropriate bitrate. Other common values are 3750000, 7500000, and 15000000.
vbr	boolean		false	When true, does not insert null packets into transport stream to fill specified bitrate. The bitrate setting acts as the maximum bitrate when vbr enabled.
dvb	boolean		false	When true, uses DVB buffer model for Dolby Digital audio. When false, the ATSC model is used.
audio_packets_per_pes	integer	>= 0	2	The number of audio packets to insert for each PES packet.
pcr_every_pes	boolean		true	When true, a Program Clock Reference value is inserted for every Packetized Elementary Stream (PES) header. This parameter is effective only when the PCR PID is the same as the video or audio elementary stream.
pcr_period	integer	0-100		Nominal time in milliseconds between Program Clock References (PCRs) inserted into the transport stream.
pat_interval	integer	10-1000	100	Millseconds between PAT tables in output.
pmt_interval	integer	10-1000	100	Millseconds between PMT tables in output.

HLS PACKAGE FILTER SETTINGS

Packages content to Apple HTTP Live Streaming. Can be applied to any content type. Endpoints can be accessed at http://<server ip>/out/i/<endpoint id>.m3u8

The HLS Package filter does not do any actual blacking out of content based on SCTE-35 messages, it assume the upstream encoder takes care of that. The blackout settings control manifest tags only.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
segment_duration	integer			Length in seconds of individual TS segments in HLS stream. Value will be rounded to the next availabe segmentation point of the source content.
index_duration	integer			Length in seconds of individual bitrate .m3u8 playlists for Live content. VOD content will emit the entire duration of content.
playlist_type	string	EVENT, VOD		Inserts EXT-X-PLAYLIST-TYPE as specified by playlist_type when packaging VOD content or downstream of a Live to VOD filter on Live content
avail_trigger	string	all, placement_opportunity		all will generate cue tags based on all SCTE-35 ad types. placement_opportunity will generate them only for placement opportunities.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
ad_markers	string	none, scte35_enhanced, passthrough		none will omit all SCTE-35 ad markers from the output. passthrough causes the output HLS manifest to contain a copy of the SCTE-35 ad markers (comments) taken directly from the input HLS manifest. scte35_enhanced generates ad markers and blackout tags based on SCTE-35 messages in the input (UDP or HLS).
broadcast_time	boolean			When true, ID3Timed Metadata messages are generated every 5 seconds with the ingest time of the content. Irrespective of this parameter, if any ID3 Timed Metadata is found in HLS input, it is passed through to HLS output.
ignore_web_delivery_allowed	boolean			When true, SCTE-35 segmentation descriptors with web_delivery_allowed_flag set to 0 will no longer trigger blackouts or avails.
ignore_no_regional_blackout	boolean			When true, SCTE-35 segmentation descriptors with no_regional_blackout set to 0 will no longer trigger blackouts or avails.
enable_blackout	boolean			When true, add blackout tags to HLS manifest based on SCTE-35 Program, Chapter, and Not Indicated segmentation descriptors with web_delivery_allowed or no_regional_blackout set to 0.
enable_network_end_blackout	boolean			When true, add blackout tags to HLS manifest based on SCTE-35 Network Start/End segmentation descriptors. Requires use of network_id (see below) which is matched against the SCTE-35 message to determine blackout state.
network_id	boolean	matches EIDR ID format (e.g., “10.XXXX/XXXX-XXXX-XXXX-XXXX-XXXX-C”)		The Network ID for use with enable_network_end_blackout (see above).
include_program_date_Time	boolean		false	Inserts EXT-X-PROGRAM-DATE-TIME tag in .m3u8 manifest files. If source content is HLS and contains EXT-X-PROGRAM-DATE-TIME tags, the source time is used. Otherwise, the time that the source segment is processed by the input filter is used.
program_date_time_interval	integer		600	Interval (in seconds) to insert EXT-X-PROGRAM-DATE-TIME tags into manifest files.

LIVE TO VOD SETTINGS

Live to VOD filters are used to turn a linear stream into a VOD asset. The start and end times are given to make a clip out of the linear stream. When a Content Window is specified in the input filter, content between the start and end times will be retained on disk when the input Content Window is passed. Live to VOD filters must be created before the content window expires in order for them to be retained. When the Content Window is set to Packager Controlled, the upstream packager must be configured to retain content to fit the Live to VOD window.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
start_time	datetime			Calendar date/time to start. Required unless allow_url_start_end_params is enabled.
end_time	datetime			Calendar date/time to end. Required unless allow_url_start_end_params is enabled.
start_over	boolean		true	When enabled, during the Live to VOD window playback will start at the beginning of th window. When disabled, playback will start at the current time. This has no effect after the end time of the Live to VOD window.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
allow_url_start_end_params	boolean		false	When enabled, URL parameters control the VOD assets returned from this filter and any child filters. Parameters "start" and "end" can be supplied as ISO 8601 dates (example: http://<web server>/out/u/filtername.m3u8?start=2015-04-27T17:30:00+00:00&end=2015-04-27T18:00:00+00:0 or POSIX time (example: http://<web server>/out/u/filtername.m3u8?start=1430155800&end=1430157600). Parameters can also be path elements of the URL for players that do no pass arguments through to child requests (example: http://<web server>/start/1430155800/end/1430157600/out/u/filtername.isr Manifest)

MSS PACKAGE FILTER SETTINGS

Packages content for Microsoft Smooth Streaming players. Can be applied to any content type. MSS Package endpoints provide the manifest file at http://<server ip>/out/i/<endpoint id>.ism/Manifest

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
fragment_duration	integer			Length of fragments to generate. Actual fragments will be rounded to the nearest multiple of source content fragment length.
index_duration	integer			Length in seconds to emit in manifest for live content. Parameter is ignored for VOD content.

PLAYREADY SETTINGS

Playready filters are used to apply Playready DRM to MSS packaged content when requested by an end user. It must be chained after an MSS Package filter. Elemental Delta is integrated with several key providers.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
keyprovider_type	Key Provider Type	custom, irdeto, conax, piksel, microsoft, vosp	custom	Playready key provider type. Custom allows the user to manually enter Playready encryption settings.
keyprovider_settings	Key Provider Settings	Irdeto Settings, Conax Settings, Piksel Settings, Microsoft Settings, VOSP Settings		Key Provider-specific settings.
initial_iv	integer		1	Initial value of IV.
key_id	string	GUID		Specifies a key ID to use for Playready DRM, must be a valid GUID.
key_seed	string	base64 encoded		Contains a base64-encoded key seed. Only required if content_key is not specified.
content_key	string	base64 encoded		Contains a base64-encoded content key. If exists, key_seed is not required and ignored.
license_url	string			Contains the URL for the license acquisition Web service.
ui_license_url	string			Contains the URL for a non-silent license acquisition Web page.
custom_attributes	string			The content author can add arbitrary custom attributes inside this element. Microsoft code does not act on any data contained inside this element.

MICROSOFT SETTINGS

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
program_id	string			The program ID is a unique identifier associated with a set of keys. It is passed to the keyprovider URI.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
uri	URL			URI used for retrieving the keys from the keyprovider.
username	string			Used to authenticate to the keyprovider.
password	string			Used to authenticate to the keyprovider.
ui_license_url	URL			Contains the URL for a non-silent license acquisition Web page.
custom_attributes	string			The content author can add arbitrary custom attributes inside this element. Microsoft code does not act on any data contained inside this element.

CUSTOM KEYPROVIDER SETTINGS

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
key_id	string	GUID		Specifies a key ID to use for Playready DRM, must be a valid GUID.
key_seed	string			Contains a base64-encoded key seed. Required unless content_key is specified.
content_key	string			Contains a base64-encoded content key. Required unless key_seed is specified.
license_url	URL			The URL for the license acquisition Web service.

TIME DELAY SETTINGS

A time delay filter introduces a buffering delay between the incoming content and its play back. Input filter content window should be set larger than the desired time delay.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
delay_seconds	integer			Number of seconds to buffer the content before being played back.

USER AGENT SETTINGS

User agent filters are used to select what user agents are considered to apply to the following elements in the filtering chain. For example, a user agent filter can be inserted before an ad replace filter, so that you can select different commercials when the same channel is viewed from a STB or a mobile device.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
user_agent_preset_ids	string	Valid User Agent Preset IDs		There needs to be at least one User Agent Preset ID in a comma separated list of IDs.

VOD CLIP SETTINGS

VOD Clip filters can clip portions of a content stream and deliver them as VOD streams. The start and end times are given relative to the beginning of the content.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
start_time	integer			Start time in seconds. Required unless allow_url_start_end_params is enabled.
end_time	integer			End time in seconds. Required unless allow_url_start_end_params is enabled.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
allow_url_start_end_params	boolean		false	When enabled, URL parameters control the VOD assets returned from this filter and any child filters. Parameters "start" and "end" can be supplied as days (d), hours (h), minutes (m), and seconds (s) from the beginning of the content in URL parameters (example: http://<web server>/out/u/filtername.m3u8?start=60s&end=5m30s) or as path elements of the URL for players that do no pass arguments through to child requests (example: http://<web server>/start/60s/end/5m30s/out/u/filtername.ism/Manifest)

USER AGENT PRESET

In the user agent filter, the selection of the user agent to be matched is made with a list of user agent presets. User agent presets can be defined with the following structure.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
name	string			Name to be referenced in user_agent_preset_ids.
regex	string			Regular Expression to define the matching pattern.

OUTPUT TEMPLATE

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
name	string			Name.
filter	Filter			One or more filters that can be chained together to form the endpoints for this output template.

CLUSTER SETTINGS

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
heartbeat_interval_seconds	integer	> 0	2	The frequency in seconds at which nodes send health notifications to each other.
drop_node_after_seconds	integer	> 0	10	Number of seconds before a node will be considered failed. If the Leader node is failed, the other node will be promoted to Leader.
take_action_time_seconds	integer	> 0	20	Number of seconds to wait after a node state transition has occurred before allowing another transition.
clustermgr_interval_seconds	integer	> 0	5	Frequency of cluster health checks.

NODE SETTINGS

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
hostname				Hostname.
ssh_password				SSH password.

VIP SETTINGS

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
address	string	Valid IPv4 Address		IP address for this VIP.

NAME	TYPE	RANGE	DEFAULT	DESCRIPTION
interface	string	Valid network interface	eth0	Network interface to attach this VIP to. The interface should exist and be active on all nodes in the cluster.
netmask	integer	0 - 33	24	Integer netmask for this VIP. This should normally match the netmask on the interface the VIP is assigned to.
broadcast	string	Valid IPv4 Address		Broadcast address for this VIP. This should normally match the broadcast address on the interface the VIP is assigned to.

SNMP INTERFACE

The Elemental Delta 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 errors.

A user can interact with the system using a variety of network management systems. Elemental Delta 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:

- 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
```

The entire Elemental Delta SNMP interface can be queried via snmpwalk as follows:

```
      localhost elemental
```

NOTE: On a system with a large database, this procedure could take some time and frequent polling can affect system performance.

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.
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

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

- Configuring Authentication
- Managing Users
- User Profile
- Authentication and REST

AUTHENTICATION

The Elemental Delta 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 Delta 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.

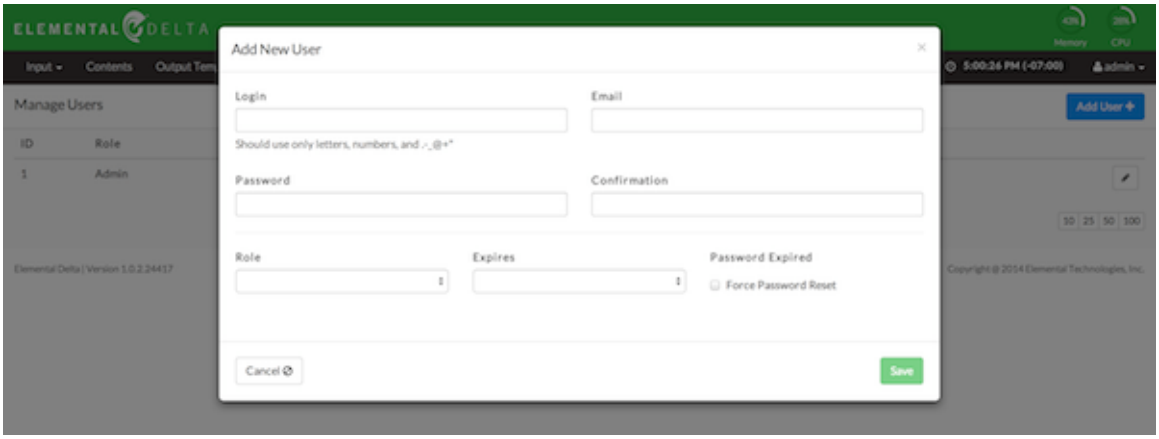
- The **Number of failed login attempts allowed** field specifies the number of login attempts allowed for a single user login before triggering a login timeout for that user login. This allows the Elemental Delta system to protect against brute-force attacks. Setting this value to 0 will disable brute-force 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 USERS

The Admin user can create and manage users on the Users page. The Users page can be reached via the user profile dropdown at the top-right of every page.

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 Expires field allows a user to be created 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 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.