ELEMENTAL® SERVER

API AND USER GUIDE

2.0.2 RELEASE

MARCH 5, 2013

TABLE OF CONTENTS

PURPOSE

This document is intended for system integrators and users of Elemental® Server. 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 Server is a powerful media transcoding solution. It can be used to convert media files from one format to many others while applying pre-processing, filtering, and scaling to format the media in many different ways.

Elemental Server 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 Server. This interface is used when a human is interacting with the server, or when no automation or integration with other systems is required.

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 Server system. It allows a management system to query the state of the service and Jobs.

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

JOBS

A transcoding Job is defined as the processing of one input media asset, or several media assets intended to be concatenated together. This includes the conversion of one set of input files to a single output file or one set of input files to many output files, including any video effects that need to be applied. See Input Codecs for a list of valid input media, and Output Codecs for a list of supported output formats.

Transcoding Jobs are prioritized, submitted, distributed, queried, and deleted through the interfaces described in this document. Elemental Server's prioritization policy balances the Job queue load on a single node or in a cluster of nodes. See the section on prioritization in this document for more details.

Job can be in the following states: PENDING, PREPROCESSING, RUNNING, POSTPROCESSING, COMPLETE, CANCELLED or ERROR. When a Job is PENDING, it has been added to the queue for processing. When it is RUNNING, a node has accepted the Job and the Job is currently being processed. The percentage complete is updated periodically by the processing node and can be queried by the user at any time. Once all outputs have been created and transferred to the desired destination, the Job goes to the COMPLETE state and performance metrics are logged. Cancelling a Job in the RUNNING or SUSPENDED state will stop the processing and remove the Job from the queue. See the Advanced pre and post processing section for details on pre and post processing

When a Job is added to the system, it must first be validated before going in to the Job queue. Validation includes ensuring the request is not malformed, and that enough information is given to process the Job. If the request fails, then the Job is not added to the queue. A second validation also validates that the input source can be found and is accessible, and is a supported video format. If the Job fails this validation, the Job will be added to the Job queue, but in the ERROR state. More details are available when a Job is in the ERROR state.

Jobs must be prioritized using the priority field when creating the Job. This field ranges from 1 to 100, with 100 being the highest priority.

JOBS SEARCH

Jobs listed in the Job Control panel can be searched using the search field located on the header of each page. A list of data associated with a Job that is searched is listed below:

- Job ID
- Job User Data
- Job Input URI
- Job Destination URI
- Job Node Hostname
- Job Created At Date*

The following date formats are supported:

- MM/DD/YYYY
- MM/DD/YY
- MM-DD-YYYY
- MM-DD-YY

* Please note that dates are stored in UTC time, not in the user's timezone. For example, if the user created a Job at 7:00 PM PDT on 11/01/2009, the Job's created date is 2:00 AM UTC on 11/02/2009.

PRESETS

A Preset is a predefined group of settings for a single output media asset. A Preset allows the user to create output files targeted at a particular device or standard output format. For example, the iPhone Preset produces media files with settings appropriate for playback on an Apple iPhone. Elemental maintains a list of common Presets that are delivered to the Elemental Server system via software updates. Additionally, the user can specify named Presets using any of the interfaces to the Elemental Server system.

JOB PROFILES

A Job Profile is a saved Job definition that includes all settings except the input media. Job Profiles can optionally contain a preroll input file and/or a postroll input file. Jobs can be submitted with input parameters and a Job Profile ID to re-use previously entered settings. Note that if a Job Profile is edited, those changes are only applied to Jobs created after the change. Jobs already in the queue will retain the settings with which they were submitted.

Some example Job Profiles are supplied by default in each release of the Elemental Server software. These examples should be copied if they are intended to be used in an actual workflow as they may change from release to release.

JOB WATCH FOLDERS

Job Watch Folders are folders accessible to Elemental Server and associated with a specified Job Profile. The folders are monitored by Elemental Server for new input media files. When a new file is placed in a folder that is associated with a Job Watch Folder, Elemental Server will automatically submit a Job with the associated Job Profile.

Some example Job Watch Folders are given by default in each release of the software. These examples should be copied if they are intended to be used in an actual workflow as they may change from release to release.

NOTIFICATION

Users can set up a Job so that a notification is sent when the Job is started, completed, generates an alert, or fails. The user can be notified in the following ways:

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

The user may also request details about any Job's status at any time. These details are described later in this document.

STATISTICS

Elemental Server is continuously logging statistics about media type, quality, speed, temperature (CPU and GPU), fan speed, and resource utilization (CPU, GPU, network, disk and memory). Historical statistics are available in the web interface, on the Stats page.

SERVER CLUSTER CONFIGURATION

Elemental Servers can be clustered together and managed as a single entity. In this case, servers are configured in a high-availability, self load-balancing configuration. Nodes can be accessed through the cluster management interfaces or individually.

Nodes can be added to the cluster using the configure script. To set up a worker node, simply select Yes when prompted if this node should be added to a cluster of Elemental servers, and provide the IP address of the cluster's management node. Nodes can be reverted to standalone mode by re-running the configuration script and selecting No when prompted with the clustering question.

ADVANCED PRE AND POST PROCESSING

Most workflows have a certain number of custom commands that must be executed before or after a Job is run. Examples of these operations include:

- Moving the input file from a device or remote location that is not achievable through the standard Job setup
- Running custom validation on input or output files before or after a conversion
- Running custom notifications before or after the Job is run

Some of these commands are supported natively through the Elemental Server user interface, and the rest can be run through custom scripts that the user provides.

JOB PRE-PROCESSING

Job pre-processing features that are built into the system include:

- **Copy to local:** This feature will copy the file from the given input URI to the local file system of the unit before the Job begins. The location on the local unit is specified in the Advanced settings. The default is /data/local_sources.

JOB POST-PROCESSING

Job post-processing features that are built in to the system include:

- **Delete source:** Enabling this feature will cause the input file to be removed once the Job is complete. NOTE: preroll and postroll inputs will not be deleted with this option.
- **Move source to:** Setting a directory in this field will move the file from the input URI to a given destination URI. NOTE: preroll and postroll inputs will not be moved with this option, and this feature cannot be used with the Delete Source feature.

CUSTOM SCRIPTS

For each Job created, the user can specify a pre and/or a post script to run. The user specifies a location for the script as part of the Job web interface or REST API. This location must be accessible by the server. It is recommended to put these scripts in the **/opt/elemental_se/web/public/script** directory; the Browse button for scripts is set up to search this directory. **/opt/elemental_se/web/public/script/example_script.rb** is an example script that parses the input parameters using Ruby and prints them to the sequencer.output log file.

The pre processing script is called from the elemental_se service just before the Job runs and must have execute permission for the elemental user. The Job's state is changed to PREPROCESSING when the pre script is running, and POSTPROCESSING when the post script is running. The Job can still be cancelled when it is in one of these states. The reported start and end times for a Job will contain the running time of these scripts; however, the elapsed time only measures the time spent processing video.

The script is passed a JSON-formatted hash. The overall structure is described below:

- **id** Job ID
- **script_type** PRE for preprocessing, POST for post processing
- **inputs** Array of all inputs. Each item in the array contains the following keys:
  - **type** Type of input (file_input)
  - **uri** Path to input file
- **output_groups** Array of all output groups. Each item in the array contains the following keys:
  - **name** Indicates group type (file, Apple HLS, Adobe HDS, MS Smooth, etc.)
  - **outputs** Array of all outputs in this group. Each item in the array contains the following keys:
    - **output_path** Contains the path of the output destination
    - **video** Hash of basic video settings. Contains the following keys:
      - bitrate
      - height
      - width
      - codec
    - **audio** Array of audio streams in this output. Each audio stream in the array is a hash of basic audio settings containing the following keys:
      - bitrate
      - sample rate
      - codec

The set of destination information is sent out one for each output.

The script should return 0 for success, 1 for error, 2 for warning. If the script echoes "RETURN MESSAGE:<some message>" to STDOUT then this message is inserted in the database for the Job. Only one message can be sent back to the system and stored with the Job in the database; however, all messages and outputs to STDOUT will be present in the sequencer log file. Errors will not allow the Job to continue, but warnings will.

Some very useful tools are included with this product to help run some of the pre and post processing scripts. They are located in the /bin directory under the installation directory and include:

- **ffmpeg:** a universal video processing utility
- **mp4box:** an MP4 muxing and demuxing utility
- **ldecod:** the reference H.264 decoder from the JM group

Most linux tools are available as well, including grep, awk, sed, perl, python, and ruby.

ERRORS AND WARNINGS

There are many reasons the system may log an error or a warning. Errors and warnings can be the result of processing media, the operating system, Job settings, or improper use of the REST or user interface. When an error occurs during Job processing, the system logs the error in the database, and terminates the Job. The Job is colored red in the HTML user interface to indicate an error occurred during processing and its state is set to the ERROR state. There is no way to retry the Job without duplicating the Job. When a warning occurs, the system logs the warning in the database, and the system continues the Job. The user is optionally notified via email or web callback when an error or warning occurs. In the main Job page, Jobs can be filtered based on errors.

Media processing errors can be caused by invalid inputs or destinations, permission problems, errors in the input media or some other internal error. Jobs may also fail due to operating system issues such as running out of disk space, network timeouts, or loss of connection to the data store. Depending on the issue, sometimes retrying these Jobs may allow them to complete. If there are operating system issues that need to be addressed such as running out of disk space or networking issues, these are best investigated through the operating system shell or SMB data share.

When a Job is submitted, the parameters are validated. If a validation fails, the problem areas are highlighted in red. If a Job is being submitted through the REST interface, and the validation fails, an error message is returned (See the REST interface section for more details on validation errors).

TROUBLESHOOTING

Problems with Elemental Server may be diagnosed by viewing the log files available here: http://server_ip/logs. By clicking on the "Send to Support" link on the log files page, all of the log files for the system, as well as information about the Jobs being run and the statistics from the system will be forwarded via email to Elemental support. No media content files will be sent to Elemental.

For additional support, contact your Elemental support representative, or email techsupport@elementaltechnologies.com

WEB INTERFACE QUICK START GUIDE

Elemental Server includes a web interface to help you get started transcoding quickly. This page explains the basic steps for using the default web interface and defines the terms used in the interface.

DEFINITION OF COMMON TERMS

- **Job:** A transcoding Job is defined as the processing of input assets. This includes the conversion of input(s) to a single output file or to many output files, and also includes any video effects that need to be applied.

  How to create a Job

  How to create a Job from a Job Profile

  How to create a Job using a Job Watch Folder

- **Preset:** A Preset is a predefined group of settings for a single output media asset. This includes both the encoding parameters as well as the effects to be applied.

  How to create a Preset

  How to create a Preset from an existing Job

  Note about editing Presets

- **Job Profile:** A Job Profile is a saved Job definition that includes all settings except the input media. A Job Profile may also contain groupings of Presets.

  How to create a Job Profile

- **Input:** An input contains information about the source files. A Job can define multiple inputs, for input stitching.
- **Stream:** A stream is a predefined group of video and audio encode settings for a single encoding output. This includes both the encoding parameters as well as the effects to be applied.
- **Group:** Groups contain the common information for an output delivery format. The information included in a group is different for each group but all of the information required for delivery to an output is contained in its group. For example, groups may contain encryption information. In addition, each stream may need specific information for each group, such as the file name for the output.

  How to set up an Archive Group with Outputs

  How to set up an Apple HLS Group with Outputs

  How to set up an Adobe HDS Group with Outputs

  How to set up an MS Smooth Streaming Group with Outputs

- **Output:** An output is made up of the combination of a stream and a group.
- **Job Watch Folder:** Job Watch Folders indicate a directory that is to be watched for incoming media. A specified Job Profile is then automatically used to create a new Job for each media file that enters this directory.
- **Preset Category:** User-defined category that can be used to organize Presets.

DEFINITION OF COMMON ICONS

Icons are used throughout the user interface to indicate the state of various items and actions that can be triggered. In most cases, hovering the mouse pointer over an icon will display a small tooltip that indicates its action. An example of how this looks can be found in the Job Control page screenshot.

Show:
This icon indicates that more information is available about the given object. For example, this icon is used on the Job Control page to link to detailed information about a Job.

Edit:
This icon indicates that the given object can be edited. This is used on the Presets and Profiles page. Jobs cannot be edited, nor can the default Presets that come pre-loaded with Elemental Server.

Duplicate:
This icon indicates the duplication of an object. It is used on the Job Control page, the Presets page and the Profiles page. Clicking this icon will navigate to the New page for the given object, with all of the information filled out from the duplicated object. Duplication is very useful if only small modifications are needed for a new object.

Create Job:
This icon is found on the Profiles page and is a quick way to generate a Job from a given Job Profile. Clicking this icon will navigate to the New Job page with information filled out from the given Job Profile.

Delete:
This icon allows for the deletion of objects. It can be found on the Presets, Profiles and Watch Folders pages. Note that the collection of default Presets that come loaded with Elemental Server cannot be deleted.

Cancel:
This icon is used to cancel a running Job.

Archive:
This icon is used to archive a cancelled, completed or errored Job. Archiving a Job does not delete it, but it removes it from the main Job Control page. Archived Jobs can be found by clicking the Archive filter button on the Job Control page.

WEB INTERFACE NAVIGATION

There are 7 base pages for the default web interface

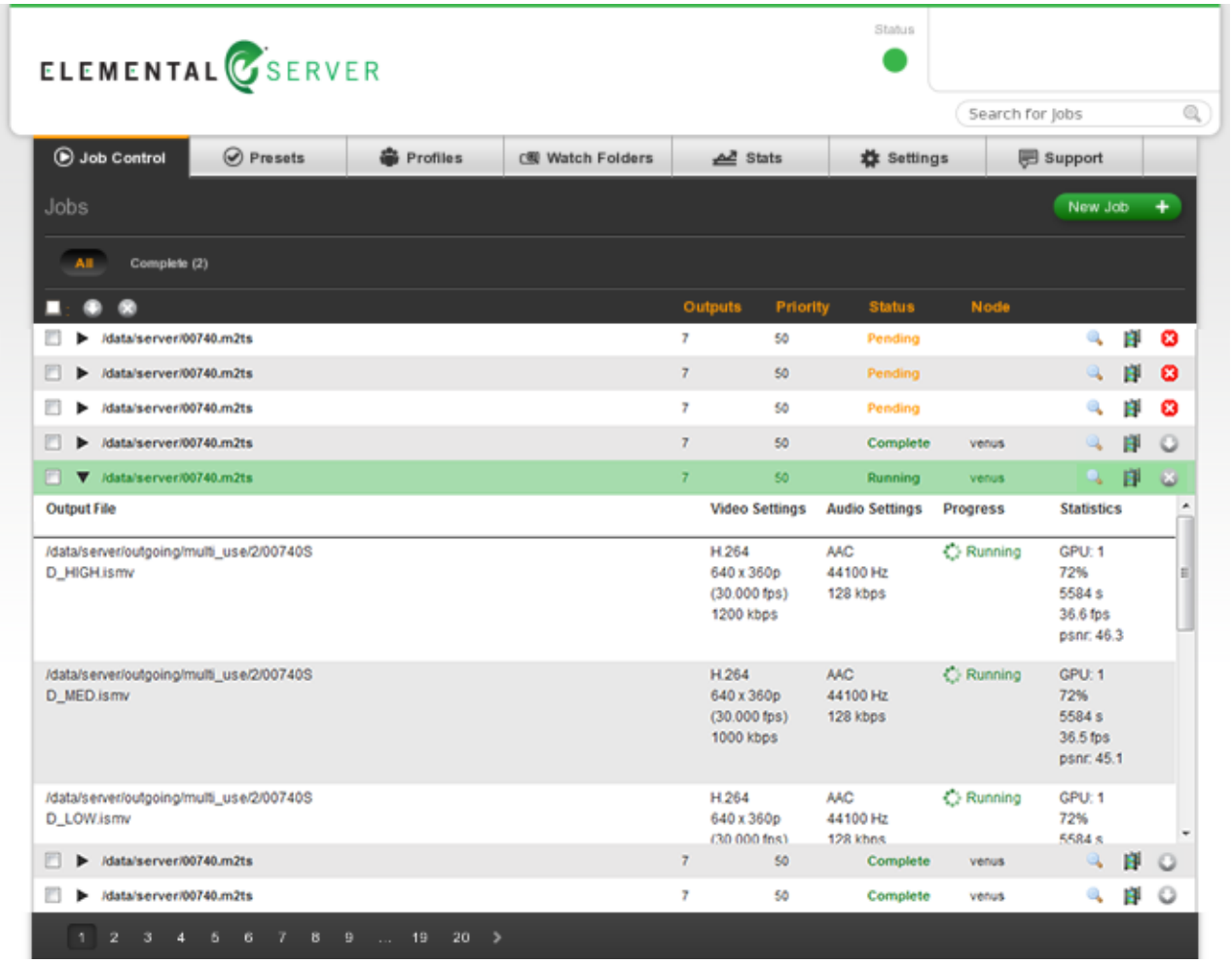
- Job Control: View status of current Jobs, current and past Job details, or create new Job
- Presets: View, create and edit Presets and Preset Categories
- Job Profiles: View, create and edit Job Profiles
- Job Watch Folders: View, create and edit Job Watch Folders
- Stats: Provides statistics for Elemental Server
- Settings: Modify Elemental Server settings
- Support: Documentation for the web interface, the REST interface, and the SNMP API

TYPICAL STEPS FOR GETTING STARTED WITH ELEMENTAL SERVER

Point a web browser at the Elemental Server web address

http://<ip address of server>

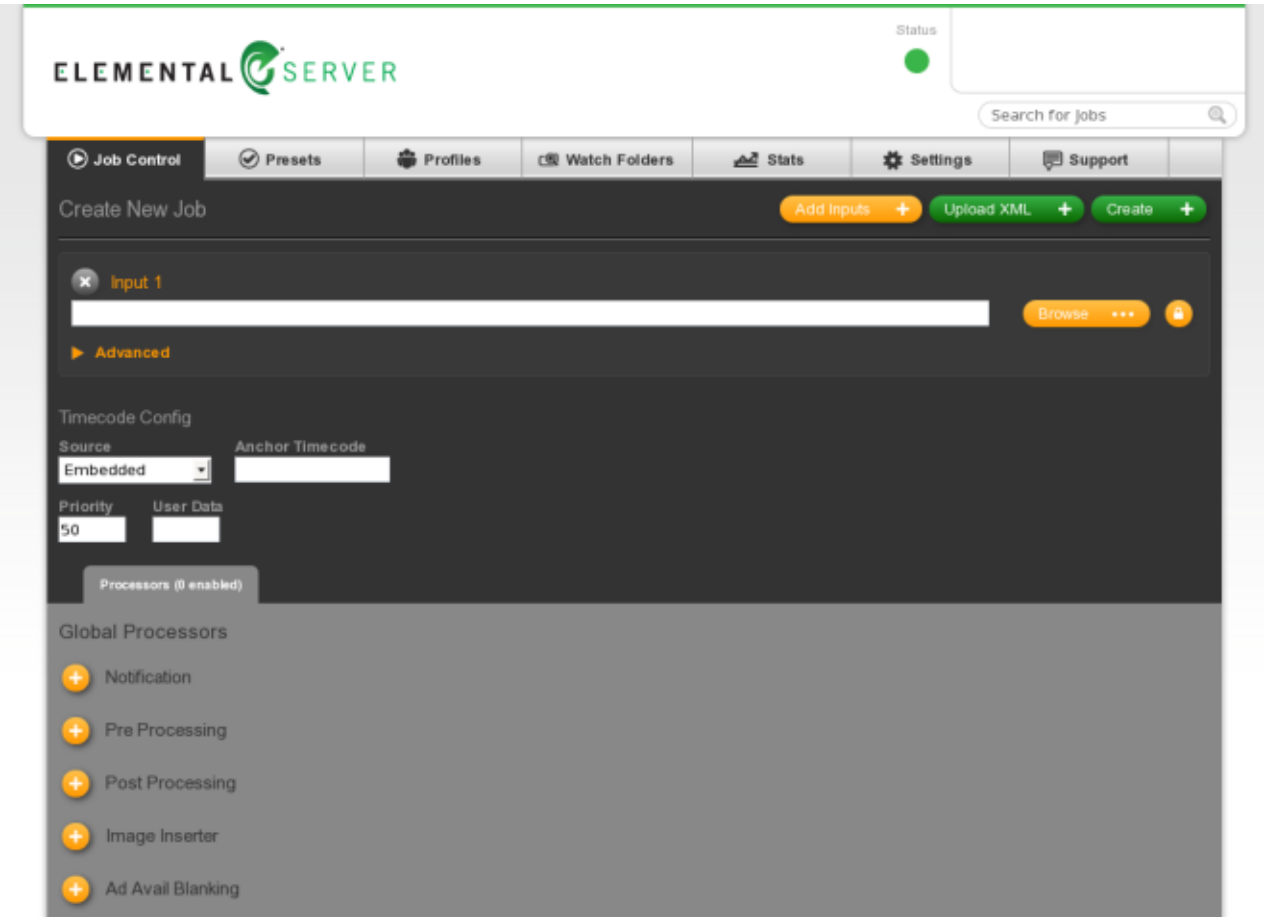
You should see a screen like this



If this is the first time the server has been started, then the Jobs list will be empty.

CREATING A NEW JOB

After selecting the *Jobs* page, simply click the "New Job" button.



To begin, configure inputs, output groups, outputs, and streams. Then set any other options you would like this Job to use.

CONFIGURING INPUTS

To configure an input, simply select a file location. Additional inputs can be added by clicking the "Add Input" button. Inputs will be concatenated together in the output file, and can be clipped by timecode under the Advanced section. See Input Clipping for more information.

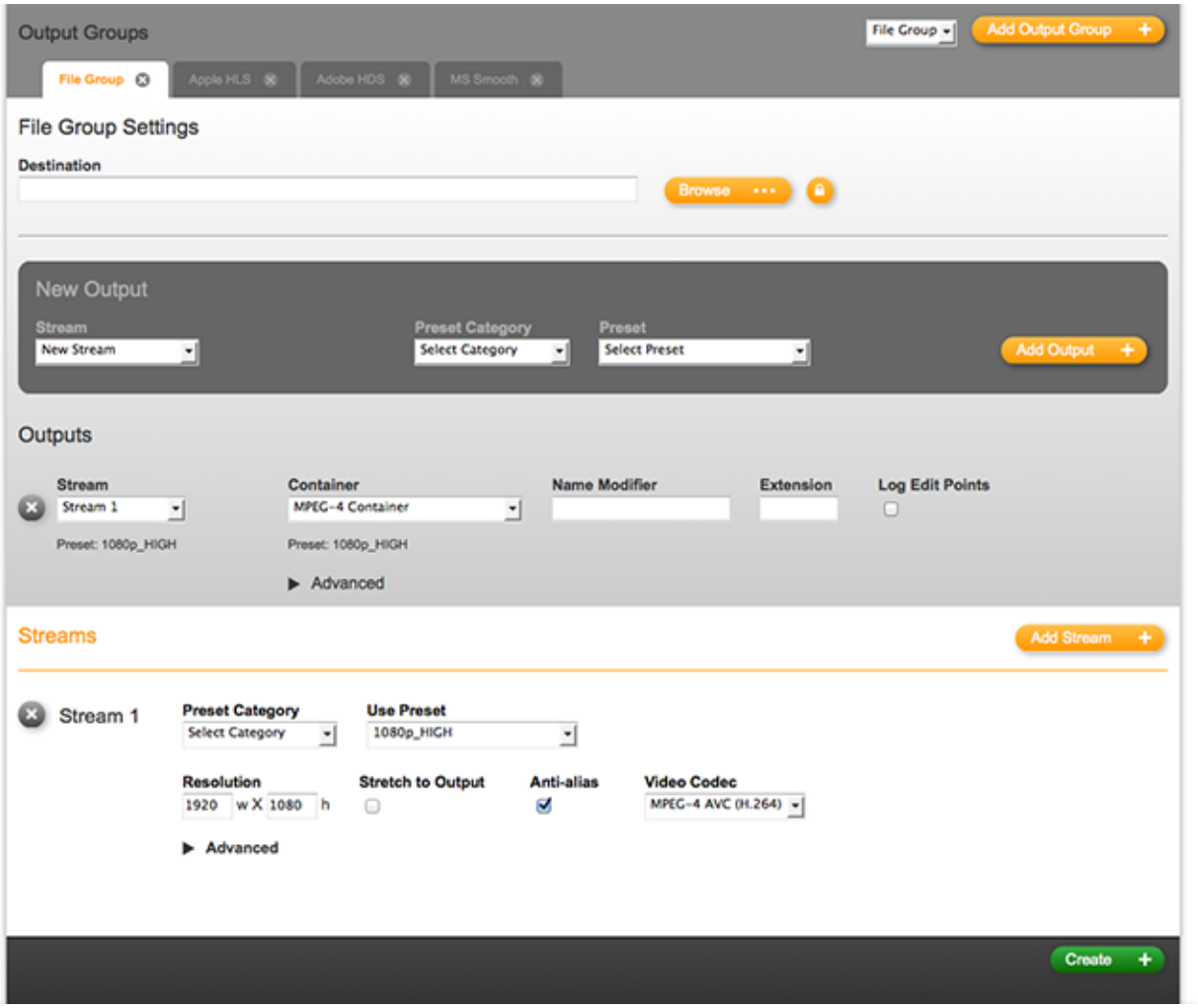
CONFIGURING OUTPUTS

Setting up an output involves configuring both a stream and an output group, in addition to the individual output. The various outputs generated by a Job are a combination of the output's stream values and its output group parameters.

First, decide on the type of outputs that will be needed for this Job (File, Apple HLS, MS Smooth, or Adobe HDS). Click on an output group tab to configure parameters that will be shared among all outputs in that group. Additional output groups can be added by clicking "Add Output Group", and those that are not needed can be deleted by clicking "Delete Group". If an output group is left unconfigured (i.e. it is not associated with any outputs and parameters are left blank), the group will be automatically removed when the Job is saved.

Outputs should be added to this group in the New Output box by selecting 'New Stream' or an existing stream (to re-use video and audio encoding parameters). A Preset can be applied to a new output being added when 'New Stream' is selected. Selecting a Preset from within the New Output box applies the Output Settings to the new output and the Stream Settings to the new Stream. The Presets available to be used in this manner are only those Presets with Output Settings that are compatible with the given output group, or Presets without Output Settings.

The order of outputs can be adjusted using the orange up and down buttons. This determines ordering in manifest files for adaptive bitrate output groups.

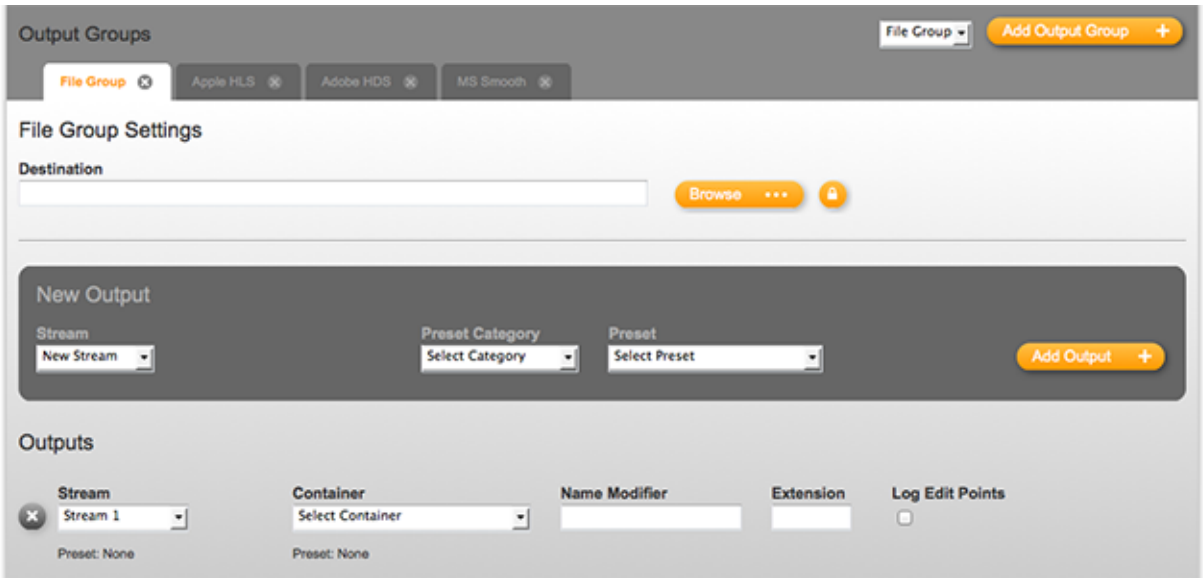


CONFIGURING STREAMS

Stream configuration can be found below the output group configuration. Streams are created when new Outputs are created; to add more simply use the "Add Stream" button. The full set of video, audio, and caption parameters are available to configure your stream by clicking the "Advanced" dropdown toggle. Each stream must be associated with at least one output among your output groups. A Preset may be selected for a stream, however only the Stream Settings from the Preset will be applied to the stream.

SETTING UP A FILE GROUP

Global File parameters found under the File tab apply to all the outputs in this group. For more detailed parameter information, see the File Group Settings parameters documentation.

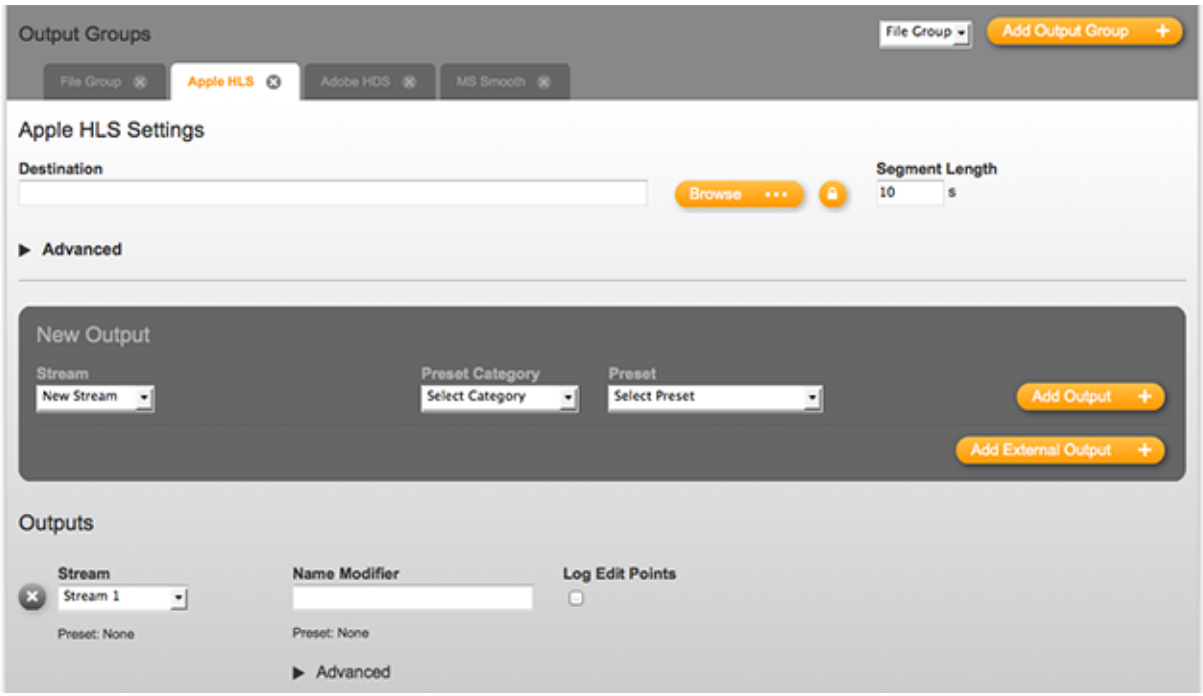


The names of outputs in a File group are a composite of the group Destination, and each output's name modifier and extension. The **Destination** field specifies the output directory and optionally a base file name. For example, setting a destination */data/server/completed/my_archive* will create files in the */data/server/completed* directory with names that start with "my_archive". Excluding the base file name by ending the destination field with a slash, for example */data/server/completed/*, will create the files in the indicated directory, and each output's filename will start with the basename of the first input.

The **Name Modifier** is appended to the information in the group destination field. For example, a name modifier of "_2400" would append to the global base file name "my_archive" in the example used previously to produce "my_archive_2400" as the final filename. Finally, the **Extension** for the output is appended to the full Destination - Name Modifier path. If no extension is specified, a default will be used based on the container.

SETTING UP AN APPLE HLS GROUP

Global Apple HLS parameters found under the Apple HLS tab apply to all the outputs in this group. For more detailed parameter information, see the Apple HLS Group Settings parameters documentation.



The **Destination** field specifies the output directory and optionally a base file name.

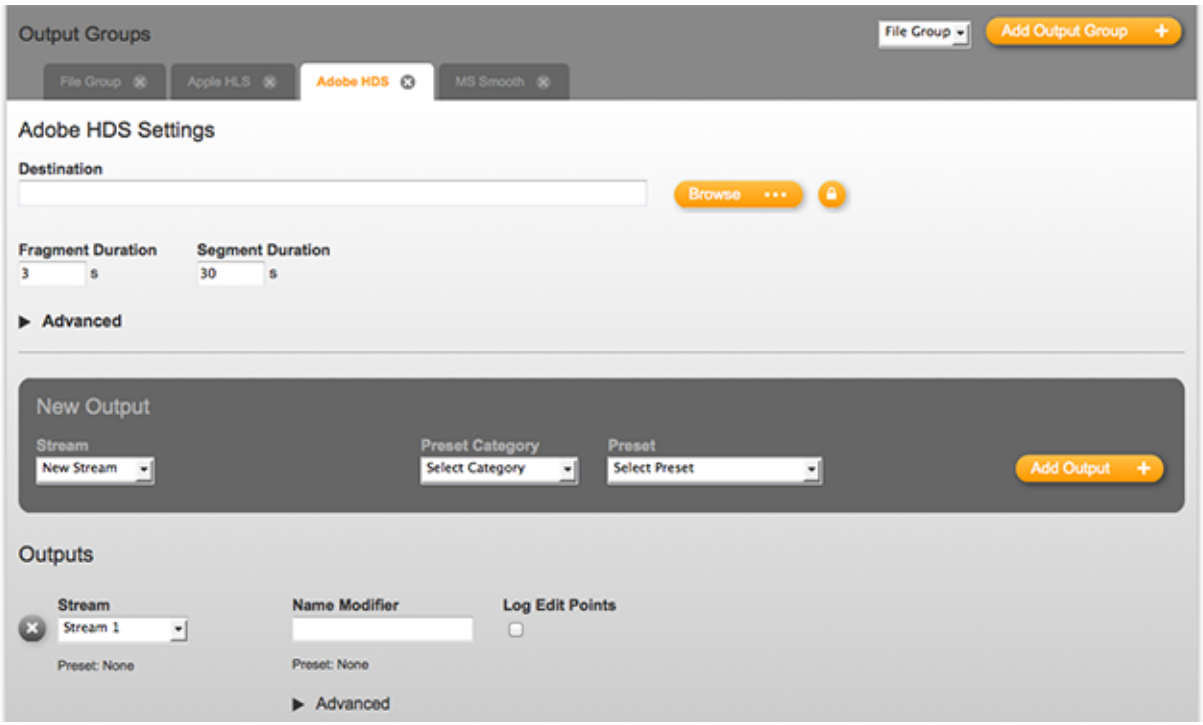
An Apple HLS group can contain two special kinds of outputs: **Audio Only** and **External** outputs.

Audio only outputs can be created by connecting an output to a stream that defines only audio settings. Audio only outputs include an advanced setting that allows you to specify a static placeholder image to embed in the output.

External outputs can be added using the "Add External Output" button. Including an external output directs the output manifest to insert an entry for an asset that is generated by a separate encoder. Note that at least one of the outputs in an Apple HLS group must have video.

SETTING UP AN ADOBE HDS GROUP

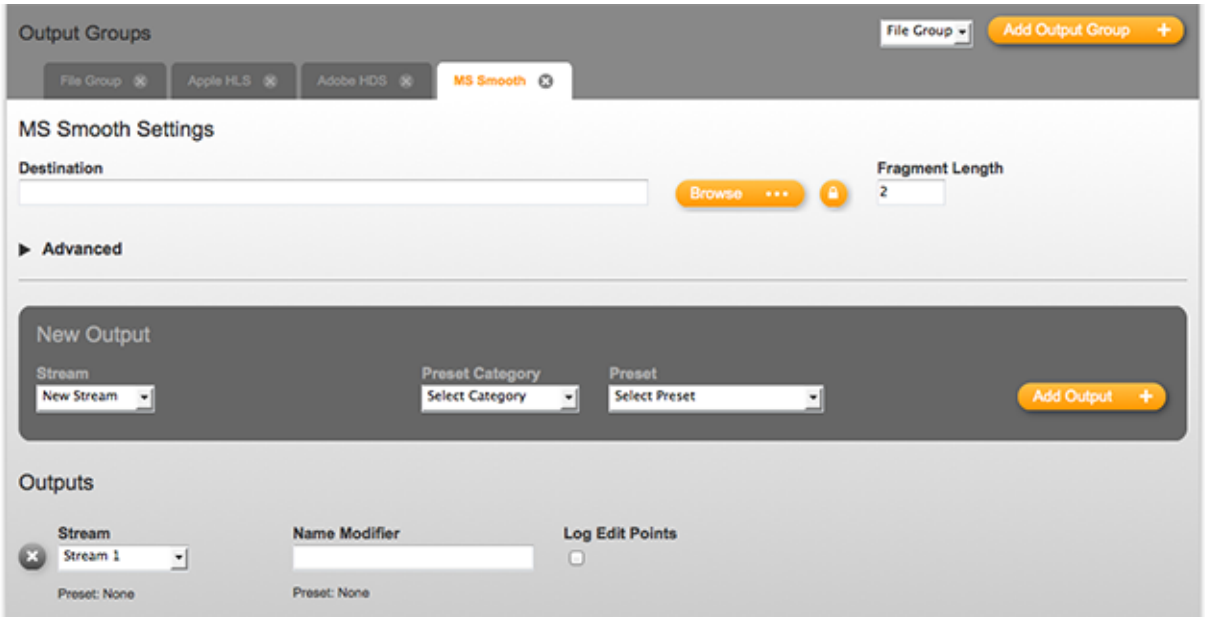
Global Adobe HDS parameters found under the Adobe HDS tab apply to all the outputs in this group. For more detailed parameter information, see the Adobe HDS Group Settings parameters documentation.



The **Destination** field specifies the output directory and optionally a base file name.

SETTING UP AN MS SMOOTH STREAMING GROUP

Global MS Smooth parameters found under the MS Smooth tab apply to all the outputs in this group. For more detailed parameter information, see the MS Smooth Group Settings parameters documentation.



The **Destination** field specifies the output directory and optionally a base file name. The **Name Modifier** is appended as usual to the information in the global destination field.

An MS Smooth group can optionally contain one or more **Caption** outputs. These can be created by connecting an output to a stream that defines only caption settings. Note that at least one of the outputs in an MS Smooth group must have video.

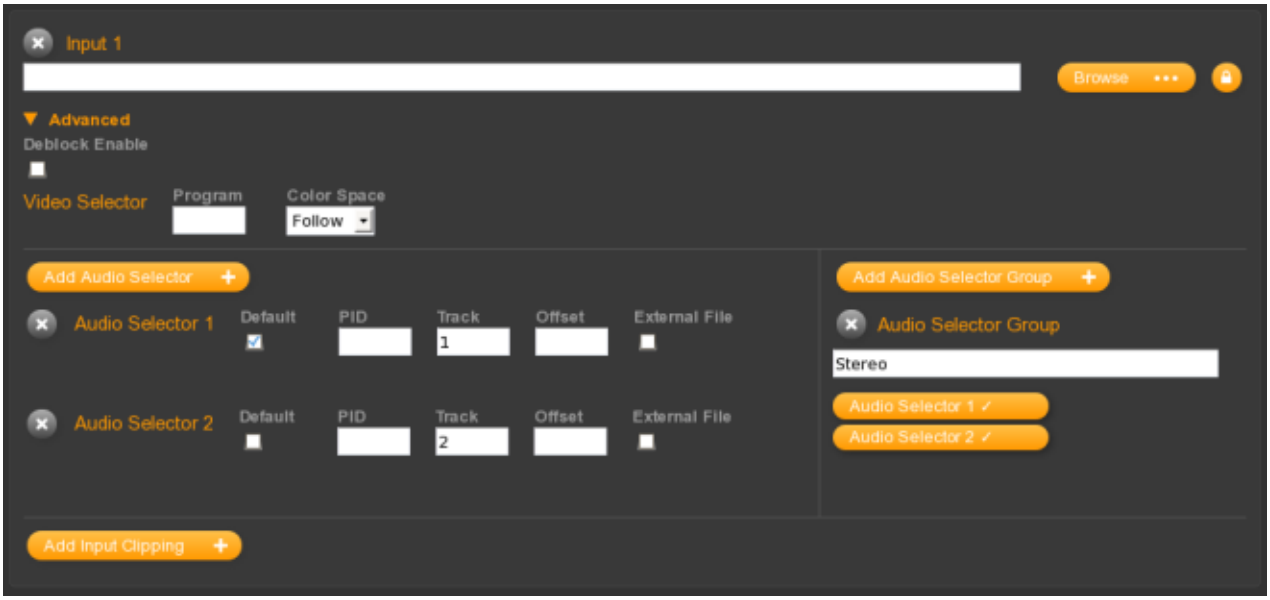
SAVING AND MANAGING A JOB

After all of your Output Groups, Streams and Outputs have been set up, click the "Create" button in the upper right to create the Job. The Job will then be placed in a "Pending" state, and the Job will begin running when the system has resources available.

The main view for a Job is called the **Control Panel**. From the "Control Panel" you can monitor a running Job, see log files, monitor the state of each output, and more. If you need more complete information about the Job, click "Details" near the top of this page. The **Details** page contains the complete set of Job parameters for reference.

ADVANCED AUDIO TRACK SELECTIONS

Elemental Server allows audio track selection from inputs with multiple audio tracks as well as tracks from external files through the use of Audio Selectors. Additionally, Selectors can be grouped to merge multiple audio tracks into a single output track. For example, to combine two mono tracks into one stereo track, add two Audio Selectors and one Audio Selector Group, select both in the group box, and name it "Stereo":



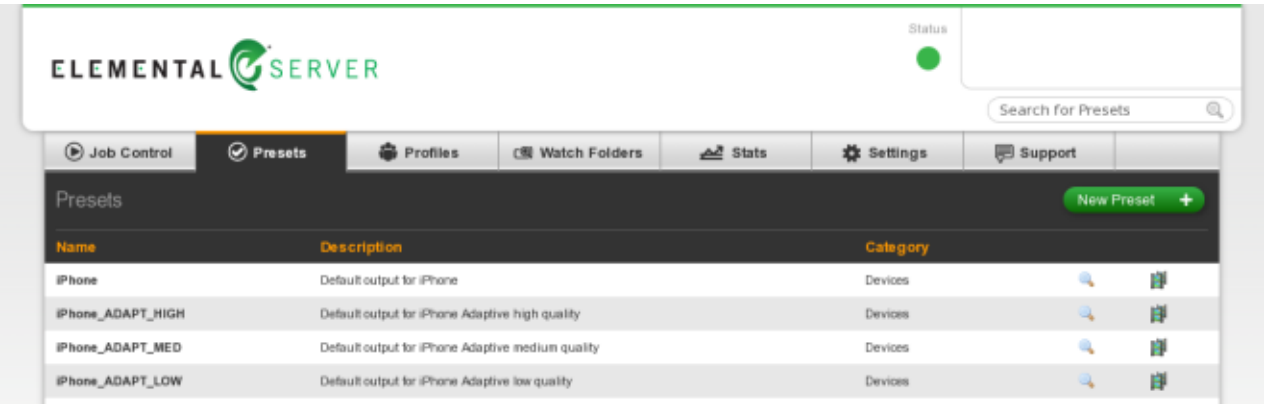
In the Audio settings of the output, note that your Audio Selectors and the "Stereo" group are available for this output.

For more information, see Audio Selector

USING PRESETS

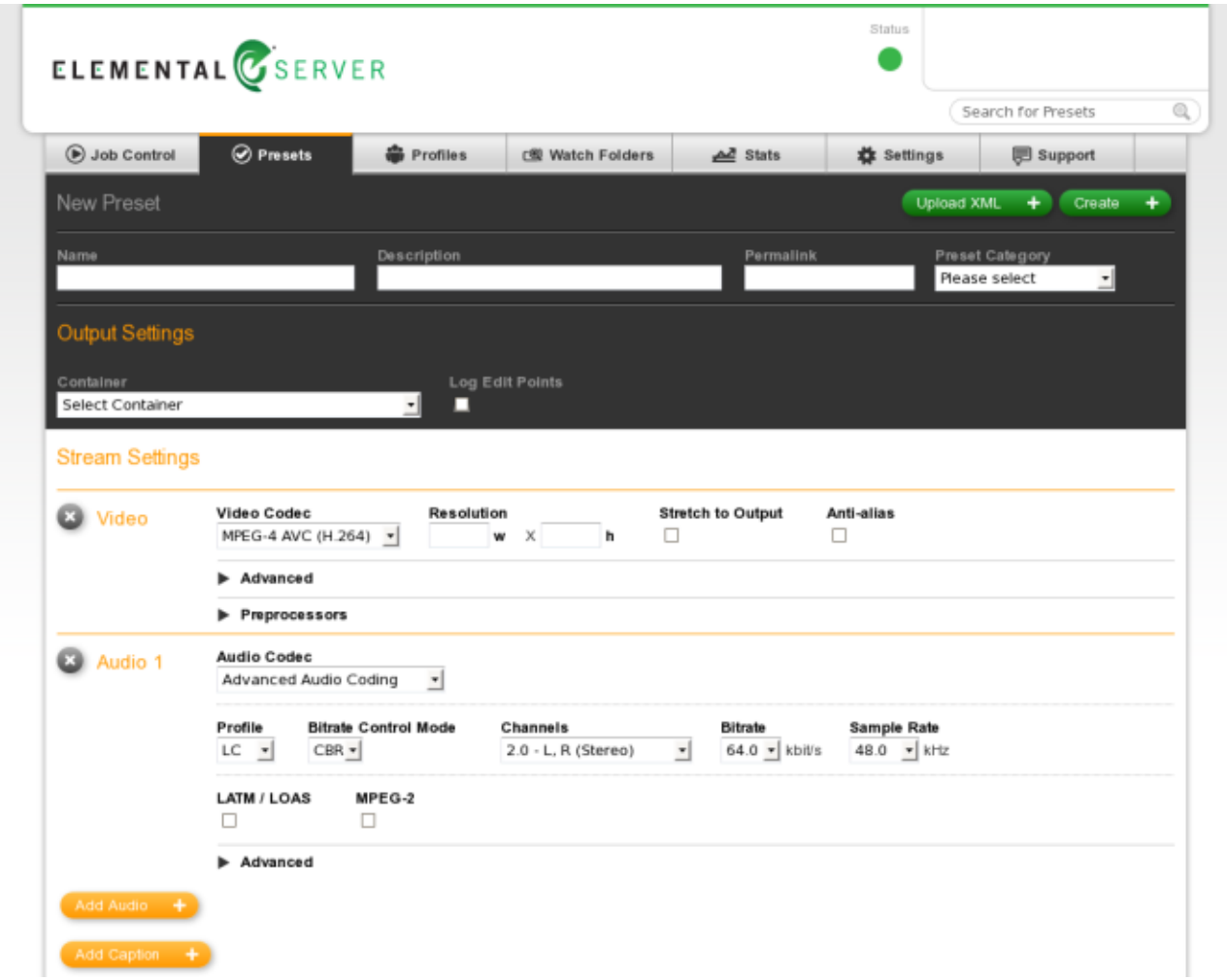
Presets simplify stream creation when the same encoding parameters will be used repeatedly.

Creating a Preset is not required, but if you plan to use the same encoding information multiple times, it is recommended. Click the *Presets* tab to view existing Presets and to create new Presets.



CREATING A NEW PRESET

Duplicate an existing Preset and modify it to meet your target settings or click the *New Preset* button. For example, duplicating the 1080p_HIGH Preset will show the detailed settings for the Preset which you can then modify to create your own custom Preset.



A Preset contains two distinct settings areas, Output Settings and Stream Settings. Output Settings contains container settings that are applied to an output. A Preset does not have to specify Output Settings; leaving the container blank will create a Preset that is agnostic to output types.

Stream Settings contains video, audio and caption encoding information that are applied to a stream. A Preset must specify encoding information.

For a more detailed description of each of the available settings please see Preset Parameters. After making your desired changes, including a new **Name** and **Description** for the Preset, select the *Save* button to commit the changes to the database.

CREATING A NEW PRESET USING AN EXISTING JOB

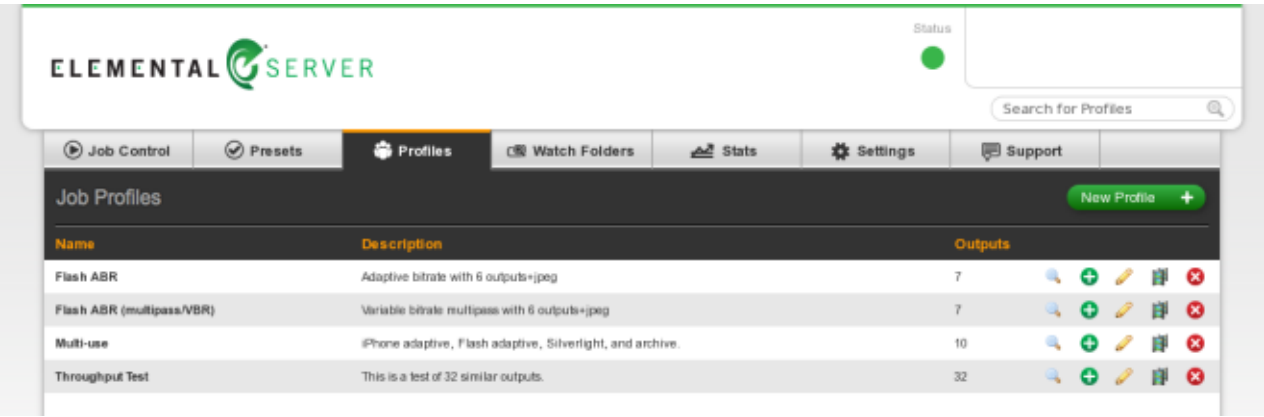
Creating a Preset using the Stream settings of an existing Job is useful if these settings will be used again. Navigate to the "Show" page for the Job and find the Stream that you wish to save as a preset. Next to the Stream label is a "Save as Preset" button. Clicking this button brings up fields for selecting the Preset Name and Description, as well as Preset Category. Clicking "Save" will save the Preset.

EDITING PRESETS

Note that whenever a Preset is edited, the Job Profiles and Jobs that had been created using this Preset will not be updated. If a Preset must be edited, all associated Job Profiles will need to be updated to use the updated Preset.

USING JOB PROFILES

Creating Job Profiles can simplify Job creation while making sure that your Jobs share the same set of stream and group options.



After navigating to the *Profiles* page, each existing Job Profile is listed. By selecting a Job Profile, you can see the details of the stream and group settings.

USING THE JOB PROFILES PAGE TO CREATE A NEW JOB

There is a Create Job icon listed for each Job Profile on the main Profiles page. Clicking this icon will take you to the New Job page with all the information from the given Job Profile already filled out. All that remains to be done is to specify the input source. Additionally, any changes to the Job Profile's parameters can be made at this time as well.

CREATING A NEW JOB PROFILE

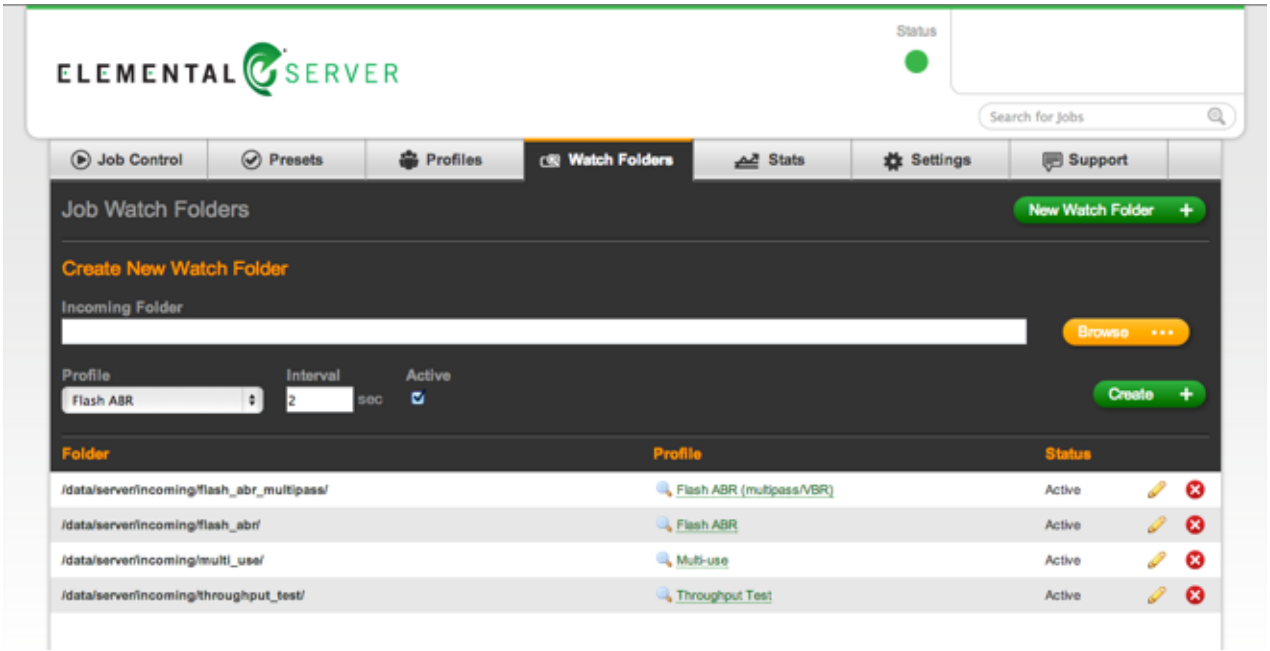
Next, create a Job Profile to contain the Preset or list of Presets along with the details of any pre- or post-processing scripts and where the output files should be stored. This can be accomplished by clicking the *Profiles* tab and then clicking the *New Profile* button. The settings for a Job Profile are very similar to the New Job page.

The top section of the detailed Job Profile page shows the information related to the Job Profile while the lower section of the page shows the outputs associated with the Job Profile. More details on the available Job Profile Parameters can be found on the Parameters page.

USING JOB WATCH FOLDER

The Job Watch Folder page allows you to create Job Watch Folders that will automatically perform a transcode anytime a file is moved into the specified incoming folder. The Job Profile determines the transcoding settings to apply and also indicates

whether the input files are moved or deleted after the transcode is complete. Job Watch Folders are ideal for automated transcodes.



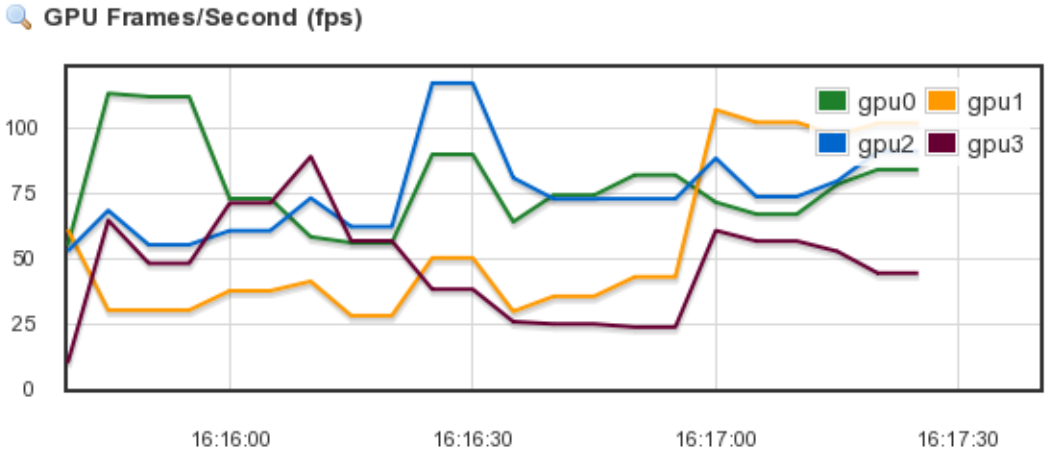
USING THE STATS PAGE

The stats pages provide various statistics about Elemental Server. In addition to providing statistics on the cluster and each individual node, there is access to the Alerting system as well as to the list of Logs provided by the system.

CHARTS

A variety of charts logging information about the system are shown on the stats pages. These charts update automatically in real-time.

To enlarge any particular chart in order to see more detailed information, simply click on the chart's title or the chart itself. An example chart showing the Frames per Second being transcoded on each GPU is shown below.



CLUSTER STATISTICS

After navigating to the Stats tab, statistics for the Elemental Server cluster are shown. This includes charts providing information on the Percent Realtime and Total Frames per Second of the currently running Jobs, as well as the current Queue Length. There is also historical information about the total number of output streams that the system has produced over various time frames.

At the bottom of the page is a list of all the nodes within the cluster. For each node you can quickly see if the node is currently active, as well as information about the number of Jobs currently running on that node, the number of Completed Jobs, and the Average Output FPS for that node. There are also charts showing the Percent CPU Utilization and the GPU Frames per Second being processed by the node.

In order to get more detailed statistics on a particular node, click on the node's name to navigate to the Node Statistics page.

NODE STATISTICS

More detailed information about a particular node can be found on the Node Statistics page, including charts providing information on the node's CPU usage, Memory usage, Disk usage, GPU temperature and GPU frames per second.

At the bottom of the page is a list of the currently running Jobs as well as the last 10 completed Jobs. Links to more detailed information about each Job are available.

USING THE SETTINGS PAGE

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

GENERAL SETTINGS

The General Settings page allows for selecting the timezone for the Elemental Server system, and also provides an option for disabling the browser warning that appears on unsupported browsers. Note that disabling the browser warning only affects the current browsing session. There are also options for managing the cluster-wide background tasks that auto-archive and auto-delete Jobs and delete old thumbnail images off each node. Additionally, settings for the Global Alert Notification are located on this page. The Global Alert Notification is a set of default notification settings that will be applied to any new alert that is created on the Elemental system.

NETWORK SETTINGS

The Network Settings page provides access to the hostname of the Elemental Server. Additionally, DNS servers can be added, edited and removed from this page. The Network Settings page also offers the ability to manage both ethernet devices, Eth0 and Eth1. Configuration of Eth1 is not required. Both ethernet devices can be configured to use DHCP, or the settings can be entered manually.

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

Default settings can be restored by clicking the "Restore Defaults" button at the top of the page. This will display the default network settings and the current hostname. Adjustments may then be made to the default settings. In order to commit these changes, the "Save" button must then be pressed.

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 Server. 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 Server. There is a checkbox available to mark any open incoming ports for deletion, and there is a field below to add a new incoming TCP or UDP port. Incoming ports must be added one at a time.

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

SNMP SETTINGS

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

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

AUTHENTICATION SETTINGS

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

ADVANCED SETTINGS

The Advanced Settings page provides access to settings for fine-tuning the video transcoding sequencer. The CPU Load Factor controls the number of available CPU threads. This value scales by default with the number of cores and their clock rates. There are also options for managing the background tasks that auto-archive Jobs, delete old thumbnail images off the machine, and auto-delete Jobs. Auto-archive will remove completed Jobs from the displayed Job list, however they will still be available for search. Auto-delete permanently removes Jobs from the system. Setting these values to "0" will disable this functionality.

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

Default settings can be restored by clicking the "Restore Defaults" button at the top of the page. This will display the default Sequencer and Tasks settings. Adjustments may then be made to the default settings. In order to commit these changes, the "Save" button must then be pressed.

CLUSTER SETTINGS

The Cluster Settings page provides access for viewing the management node and data share information for the cluster. This is only available on clustered systems. These settings are not editable from the UI -- they are only editable via the configure script.

WEB SERVICES REST INTERFACE

The Elemental Server 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). Additional headers are required when authentication is enabled on the server.

API VERSIONS

When submitting REST requests manually or from within an automation system, it is recommended to use an API version prefix for all endpoints. The API version prefix allows you to specify which API version the server should use to interpret your data. For example, POST /api/v2.0/jobs will send a request to the /jobs endpoint, and the server will interpret the data as compatible with Elemental API version 2.0. To specify the most current up-to-date API version, simply omit the version: POST /api/jobs. Responses from the server will always be formed according to the current API version.

SIMPLE EXAMPLES

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

```
curl -H "Accept: application/xml" http://server_ip/api/jobs
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<job_list>
<job href="/jobs/1" \
    version="2.0.2.xxxx" product="Elemental Server">
  <input>
    <order>1</order>
    <file_input>
      <uri>/data/incoming/test.mp4</uri>
    </file_input>
    <name>input_1</name>
    <video_selector>
      <color_space>follow</color_space>
      <order>1</order>
      <program_id nil="true"></program_id>
      <name>input_1_video_selector_0</name>
    </video_selector>
    <audio_selector>
      <default_selection>true</default_selection>
      <order>1</order>
      <name>input_1_audio_selector_1</name>
    </audio_selector>
  </input>
  <timecode_config>
    <source>zerobased</source>
  </timecode_config>
  <priority>50</priority>
  <user_data></user_data>
  <avsync_enable>true</avsync_enable>
  <submitted>2012-12-11 13:10:48 -0800</submitted>
  <status>complete</status>
  <pct_complete>100</pct_complete>
  <average_fps>56.9</average_fps>
  <start_time>2012-12-11 13:10:49 -0800</start_time>
  <complete_time>2012-12-11 13:11:12 -0800</complete_time>
  <elapsed>22</elapsed>
</job>
...
<next href="http://server_ip/jobs?page=2"/>
</job_list>
```

Adding or updating resources is accomplished by issuing an HTTP POST or PUT command with the body containing XML data that describes the resource. To create a new job watch folder:

```
curl -H "Accept: application/xml" -H "Content-type: application/xml" \
     -d @filename http://server_ip/api/job_watch_folders
```

where the file indicated by filename contains

```
<job_watch_folder>
  <incoming>
    <uri>/data/server/folder/</uri>
  </incoming>
```

```
  <profile>1</profile>
</job_watch_folder>
```

returns

```
<?xml version="1.0" encoding="UTF-8"?>
<job_watch_folder href="/job_watch_folders/5" \
    version="2.0.2.xxxx" product="Elemental Server">
  <incoming>
    <id>21</id>
    <uri>/data/server/folder/</uri>
  </incoming>
  <profile href="/job_profiles/1">Flash ABR</profile>
  <active>true</active>
</job_watch_folder>
```

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

CLEAN XML

The XML that is returned by the server from a GET request is not in the correct format for creating new objects. The GET XML contains <id> tags to uniquely specify the object and any sub-objects, and it may also contain status information that will not be accepted by the server in a POST command. Being able to query the server for XML that is in a valid format for POSTing to create new objects is very useful -- it can be used to duplicate jobs, or to slightly modify jobs, job profiles or presets. Therefore, the Elemental Server 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 job 4. Simply make the regular GET request and tack on an extra parameter clean=true at the end. This parameter can be used when making a GET request for a particular job, preset or job profile.

```
curl -H "Accept: application/xml" http://server_ip/api/jobs/4?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 job, or the file may be edited to make any necessary adjustments. The clean xml for a job can also be downloaded directly from the web interface -- just navigate to view the job, then click 'Job XML'.

ERRORS AND WARNINGS

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

```
curl -H "Accept: application/xml" -H "Content-type: application/xml" \
     -d "<job_watch_folder></job_watch_folder>" http://server_ip/api/job_watch_folders
```

returns

```
<?xml version="1.0" encoding="UTF-8"?>
<errors>
  <error>Profile can't be blank</error>
  <error>Incoming can't be blank</error>
</errors>
```

Errors and warnings for jobs are indicated by the status, warning, and error fields returned in the job status message. Errors and warnings include an error code and a message. For example:

```
curl -H "Accept: application/xml" http://server_ip/api/jobs/58/status
```

on a system that has errors in job 58 returns something like

```
<?xml version="1.0" encoding="UTF-8"?>
<job href="http://server_ip:80/jobs/58">
  <node>server_hostname</node>
  <user_data></user_data>
  <submitted>2012-12-10 02:13:32 -0800</submitted>
```

```
  <priority>50</priority>
  <status>error</status>
  <pct_complete></pct_complete>
  <average_fps>0.0</average_fps>
  <start_time>2012-12-10 02:20:43 -0800</start_time>
  <errored_time>2012-12-10 02:29:21 -0800</errored_time>
  <elapsed>517</elapsed>
  <elapsed_time_in_words>00:08:37</elapsed_time_in_words>
  <warning_messages>
    <warning>
      <code>102050</code>
      <created_at>2012-12-10T02:14:03-08:00</created_at>
      <message>Could not read media info from source.</message>
    </warning>
  </warning_messages>
  <error_messages>
    <error>
      <code>1999</code>
      <created_at>2012-12-10T02:29:21-08:00</created_at>
      <message>EME timeout detected</message>
    </error>
  </error_messages>
</job>
```

JOBS

The following table describes the REST job control interface

URL	METHOD	PARAMETERS	RETURNS	DESCRIPTION
/jobs	GET	Pagination parameters, Filter parameters	Job List Description	Retrieves a list of jobs in the system
/jobs	POST	Job Parameters	Job Description	Creates a new job
/jobs/<id>	GET		Job Description	Retrieves a specific job in the system
/jobs/<id>	DELETE			Permanently deletes a job
/jobs/<id>/status	GET		Job Status	Retrieves a summary of job <id>'s status, without detailed encoding parameters
/jobs/<id>/priority	GET		Job Priority	Retrieves job <id>'s priority
/jobs/<id>/priority	POST	<priority>value</priority>	Job Priority	Sets job <id>'s priority
/jobs/<id>/cancel	POST	<cancel></cancel>	Job Description	Cancels job <id> If the job is cancelled, the file will be terminated in a way that is playable unless it is cancelled on the first pass of a multi-pass encode, in which case the file will be in an undefined state and most likely not playable
/jobs/<id>/archive	POST	<archive></archive>	Job Description	Marks a job as 'archived'. Job will no longer appear in main /jobs list
/jobs/<id>/resubmit	POST	<resubmit></resubmit>	Job Description	Duplicates a job and resubmits it to the queue. A new job description is returned with the same parameters as job <id>

EXAMPLE XML: CREATE A JOB FROM A PROFILE

```
<?xml version="1.0" encoding="UTF-8"?>
<job>
  <input>
    <file_input>
        <uri>/data/server/elemental.mov</uri>
    </file_input>
  </input>
  <profile>1</profile>
</job>
```

EXAMPLE XML: CREATE A SIMPLE JOB USING PRESETS

```
<?xml version="1.0" encoding="UTF-8"?>
<job>
  <input>
    <file_input>
        <uri>/data/server/elemental.mov</uri>
    </file_input>
  </input>
  <priority>50</priority>
  <output_group>
    <order>1</order>
    <apple_live_group_settings>
      <destination>
        <uri>/data/server/outgoing/hls_output</uri>
      </destination>
    </apple_live_group_settings>
    <type>apple_live_group_settings</type>
    <output>
      <stream_assembly_name>stream_1</stream_assembly_name>
      <name_modifier>_high</name_modifier>
      <order>1</order>
      <preset>2</preset>
    </output>
    <output>
      <stream_assembly_name>stream_2</stream_assembly_name>
      <name_modifier>_low</name_modifier>
      <order>2</order>
      <preset>4</preset>
    </output>
  </output_group>
  <stream_assembly>
    <name>stream_1</name>
    <preset>2</preset>
  </stream_assembly>
  <stream_assembly>
    <name>stream_2</name>
    <preset>4</preset>
  </stream_assembly>
</job>
```

EXAMPLE XML: CREATE A MORE ADVANCED JOB

```
<?xml version="1.0" encoding="UTF-8"?>
<job>
  <input>
    <loop>false</loop>
    <order>1</order>
```

```
  <file_input>
    <uri>/data/server/elemental.mov</uri>
  </file_input>
  <name>input_1</name>
</input>
<timecode_config>
  <source>embedded</source>
</timecode_config>
<priority>50</priority>
<user_data></user_data>
<avsync_enable>true</avsync_enable>
<stream_assembly>
  <name>stream_assembly_0</name>
  <video_description>
    <afd_signaling>None</afd_signaling>
    <anti_alias>true</anti_alias>
    <drop_frame_timecode>true</drop_frame_timecode>
    <fixed_afd nil="true"></fixed_afd>
    <height>540</height>
    <insert_color_metadata>false</insert_color_metadata>
    <respond_to_afd>None</respond_to_afd>
    <selected_gpu nil="true"></selected_gpu>
    <stretch_to_output>false</stretch_to_output>
    <timecode_passthrough>false</timecode_passthrough>
    <vbi_passthrough>false</vbi_passthrough>
    <width>960</width>
    <h264_settings>
      <adaptive_quantization>medium</adaptive_quantization>
      <bitrate>1800000</bitrate>
      <buf_fill_pct nil="true"></buf_fill_pct>
      <buf_size>3600000</buf_size>
      <cabac>false</cabac>
      <flicker_reduction>off</flicker_reduction>
      <force_field_pictures>false</force_field_pictures>
      <framerate_denominator>1001</framerate_denominator>
      <framerate_follow_source>false</framerate_follow_source>
      <framerate_numerator>30000</framerate_numerator>
      <gop_closed_cadence>1</gop_closed_cadence>
      <gop_num_b_frames>0</gop_num_b_frames>
      <gop_size>90</gop_size>
      <interpolate_frc>false</interpolate_frc>
      <look_ahead_rate_control>medium</look_ahead_rate_control>
      <max_bitrate nil="true"></max_bitrate>
      <max_qp nil="true"></max_qp>
      <min_qp nil="true"></min_qp>
      <num_ref_frames>1</num_ref_frames>
      <par_denominator>1</par_denominator>
      <par_follow_source>false</par_follow_source>
      <par_numerator>1</par_numerator>
      <passes>1</passes>
      <qp nil="true"></qp>
      <qp_step nil="true"></qp_step>
      <scd>true</scd>
      <slices>1</slices>
      <slow_pal>false</slow_pal>
      <telecine>None</telecine>
      <level>3.1</level>
      <profile>Main</profile>
      <rate_control_mode>CBR</rate_control_mode>
      <gop_mode>fixed</gop_mode>
      <interlace_mode>progressive</interlace_mode>
```

```
    </h264_settings>
    <codec>h.264</codec>
    <video_preprocessors>
      <deinterlacer>
        <algorithm>interpolate</algorithm>
        <deinterlace_mode>Deinterlace</deinterlace_mode>
      </deinterlacer>
    </video_preprocessors>
  </video_description>
  <audio_description>
    <language_code nil="true"></language_code>
    <order>1</order>
    <stream_name nil="true"></stream_name>
    <aac_settings>
      <bitrate>64000</bitrate>
      <channels>2</channels>
      <latm_loas>false</latm_loas>
      <mpeg2>false</mpeg2>
      <sample_rate>44100</sample_rate>
      <profile>HEV1</profile>
      <rate_control_mode>CBR</rate_control_mode>
    </aac_settings>
    <codec>aac</codec>
  </audio_description>
</stream_assembly>
<stream_assembly>
  <name>stream_assembly_1</name>
  <video_description>
    <afd_signaling>None</afd_signaling>
    <anti_alias>true</anti_alias>
    <drop_frame_timecode>true</drop_frame_timecode>
    <fixed_afd nil="true"></fixed_afd>
    <height>234</height>
    <insert_color_metadata>false</insert_color_metadata>
    <respond_to_afd>None</respond_to_afd>
    <selected_gpu nil="true"></selected_gpu>
    <stretch_to_output>false</stretch_to_output>
    <timecode_passthrough>false</timecode_passthrough>
    <vbi_passthrough>false</vbi_passthrough>
    <width>416</width>
    <h264_settings>
      <adaptive_quantization>high</adaptive_quantization>
      <bitrate>200000</bitrate>
      <buf_fill_pct nil="true"></buf_fill_pct>
      <buf_size>400000</buf_size>
      <cabac>false</cabac>
      <flicker_reduction>off</flicker_reduction>
      <force_field_pictures>false</force_field_pictures>
      <framerate_denominator>1001</framerate_denominator>
      <framerate_follow_source>false</framerate_follow_source>
      <framerate_numerator>15000</framerate_numerator>
      <gop_closed_cadence>1</gop_closed_cadence>
      <gop_num_b_frames>0</gop_num_b_frames>
      <gop_size>45</gop_size>
      <interpolate_frc>false</interpolate_frc>
      <look_ahead_rate_control>medium</look_ahead_rate_control>
      <max_bitrate nil="true"></max_bitrate>
      <max_qp nil="true"></max_qp>
      <min_qp nil="true"></min_qp>
      <num_ref_frames>1</num_ref_frames>
      <par_denominator>1</par_denominator>
```

```
      <par_follow_source>false</par_follow_source>
      <par_numerator>1</par_numerator>
      <passes>1</passes>
      <qp nil="true"></qp>
      <qp_step nil="true"></qp_step>
      <scd>true</scd>
      <slices>1</slices>
      <slow_pal>false</slow_pal>
      <telecine>None</telecine>
      <level>3</level>
      <profile>Baseline</profile>
      <rate_control_mode>CBR</rate_control_mode>
      <gop_mode>fixed</gop_mode>
      <interlace_mode>progressive</interlace_mode>
    </h264_settings>
    <codec>h.264</codec>
    <video_preprocessors>
      <deinterlacer>
        <algorithm>interpolate</algorithm>
        <deinterlace_mode>Deinterlace</deinterlace_mode>
      </deinterlacer>
    </video_preprocessors>
  </video_description>
  <audio_description>
    <language_code nil="true"></language_code>
    <order>1</order>
    <stream_name nil="true"></stream_name>
    <aac_settings>
      <bitrate>64000</bitrate>
      <channels>1</channels>
      <latm_loas>false</latm_loas>
      <mpeg2>false</mpeg2>
      <sample_rate>44100</sample_rate>
      <profile>HEV1</profile>
      <rate_control_mode>CBR</rate_control_mode>
    </aac_settings>
    <codec>aac</codec>
  </audio_description>
</stream_assembly>
<output_group>
  <name nil="true"></name>
  <order>1</order>
  <apple_live_group_settings>
    <base_url nil="true"></base_url>
    <floating_point_manifest>true</floating_point_manifest>
    <follow_segment>false</follow_segment>
    <generate_meta_file>true</generate_meta_file>
    <index_n_segments>10</index_n_segments>
    <keep_segments>21</keep_segments>
    <restart_delay>0</restart_delay>
    <segment_length>10</segment_length>
    <segments_per_subdirectory nil="true"></segments_per_subdirectory>
    <token nil="true"></token>
    <use_subdirectories>false</use_subdirectories>
    <vod_mode>true</vod_mode>
    <destination>
      <uri>/data/server/outgoing/hls_output</uri>
    </destination>
  </apple_live_group_settings>
  <type>apple_live_group_settings</type>
  <output>
```

```
    <description nil="true"></description>
    <extension>m3u8</extension>
    <id>61</id>
    <insert_timed_metadata>false</insert_timed_metadata>
    <log_edit_points>false</log_edit_points>
    <name_modifier>_high</name_modifier>
    <nielsen_id3_passthrough>false</nielsen_id3_passthrough>
    <order>1</order>
    <output_subdirectory nil="true"></output_subdirectory>
    <preset_id nil="true"></preset_id>
    <scte35_passthrough>false</scte35_passthrough>
    <container>m3u8</container>
    <apple_live_settings>
      <alternate_audio_track>false</alternate_audio_track>
    </apple_live_settings>
    <m3u8_settings>
      <audio_packets_per_pes>16</audio_packets_per_pes>
      <audio_pid>482</audio_pid>
      <pcr_every_pes>true</pcr_every_pes>
      <pcr_pid nil="true"></pcr_pid>
      <pmt_pid>480</pmt_pid>
      <private_metadata_pid nil="true"></private_metadata_pid>
      <program_num nil="true"></program_num>
      <psi_repeat_rate>0.0</psi_repeat_rate>
      <scte35_pid nil="true"></scte35_pid>
      <timed_metadata_pid nil="true"></timed_metadata_pid>
      <transport_stream_id nil="true"></transport_stream_id>
      <video_pid>481</video_pid>
    </m3u8_settings>
    <stream_assembly_name>stream_assembly_0</stream_assembly_name>
  </output>
  <output>
    <description nil="true"></description>
    <extension>m3u8</extension>
    <id>62</id>
    <insert_timed_metadata>false</insert_timed_metadata>
    <log_edit_points>false</log_edit_points>
    <name_modifier>_low</name_modifier>
    <nielsen_id3_passthrough>false</nielsen_id3_passthrough>
    <order>2</order>
    <output_subdirectory nil="true"></output_subdirectory>
    <preset_id nil="true"></preset_id>
    <scte35_passthrough>false</scte35_passthrough>
    <container>m3u8</container>
    <apple_live_settings>
      <alternate_audio_track>false</alternate_audio_track>
    </apple_live_settings>
    <m3u8_settings>
      <audio_packets_per_pes>16</audio_packets_per_pes>
      <audio_pid>482</audio_pid>
      <pcr_every_pes>true</pcr_every_pes>
      <pcr_pid nil="true"></pcr_pid>
      <pmt_pid>480</pmt_pid>
      <private_metadata_pid nil="true"></private_metadata_pid>
      <program_num nil="true"></program_num>
      <psi_repeat_rate>0.0</psi_repeat_rate>
      <scte35_pid nil="true"></scte35_pid>
      <timed_metadata_pid nil="true"></timed_metadata_pid>
      <transport_stream_id nil="true"></transport_stream_id>
      <video_pid>481</video_pid>
    </m3u8_settings>
```

```
      <stream_assembly_name>stream_assembly_1</stream_assembly_name>
    </output>
  </output_group>
</job>
```

EXAMPLE XML: OVERRIDING JOB PROFILE OUTPUT GROUP SETTINGS.

When using a Job Profile to create a Job it is useful to override some of the fields in its output groups to customize them for the specifics of your Job. The most common use of this is to set the destination URI for each output group. Other settings within the output group can also be overridden with a few restrictions. No additional output groups can be added, and for each existing output group in the Job Profile -- the group type cannot be changed and keyprovider settings within the group can neither be added nor their type changed. Note, When overriding an output group from a Job Profile it is important to include the "order" within the output group tag to refer to a specific output group in the Job Profile. Below is an example job XML that shows how to override the destination URI of an output group from a Job Profile:

```
<?xml version="1.0" encoding="UTF-8"?>
<job>
  <input>
    <file_input>
      <uri>/data/server/elemental.mov</uri>
    </file_input>
  </input>
  <output_group>
    <order>1</order>
    <name>ms_smooth_stream</name>
    <ms_smooth_group_settings>
      <destination>
        <uri>/data/server/outgoing/new_output</uri>
      </destination>
    </ms_smooth_group_settings>
  </output_group>
  <profile>5</profile>
</job>
```

EXAMPLE XML: CREATE A JOB FROM A JOB PROFILE WITH DRM OVERRIDES.

Job profiles created with outputs using a Microsoft Smooth Streaming container and the Playready DRM system can have parts of their authentication left empty to be overriden when creating a job.

These fields include;

- Key Id (key_id)
- Key Seed (key_seed)

```
<?xml version="1.0" encoding="UTF-8"?>
<job>
  <input>
    <file_input>
        <uri>/data/server/elemental.mov</uri>
    </file_input>
  </input>
  <output_group>
    <order>1</order>
    <name>ms_smooth_stream</name>
    <ms_smooth_group_settings>
      <drm_system>playready</drm_system>
      <key_id>79e5c8f7-0c29-4bb8-9d05-f58d5d00a805</key_id>
      <key_seed>2zm1nEvbhn5v4BeItuPduw==</key_seed>
    </ms_smooth_group_settings>
  </output_group>
```

```
  <profile>5</profile>
</job>
```

If no key_id or key_seed nodes are provided, values for these parameters will be auto-generated and saved with the job.

EXAMPLE XML: USING A JOB PROFILE TO CREATE A NEW JOB WITH ADVANCED OVERRIDES

When creating a new Job using an existing Job Profile, it is sometimes necessary to override specific settings deep within the Job Profile to suit the particular needs of your Job. Some common examples of this are to override the DRM settings within an MS Smooth Group, or to update individual settings in a stream video description. This can be accomplished using a simple workflow. First, retrieve the 'clean' XML for the Job Profile you want to use. This can be done via a REST request, or using the web interface. Second, the XML must be modified to transform it into a valid Job XML. Any specific fields within the XML can then be overridden. Finally, submit the modified XML via REST or the web interface to create your new Job. Consider the following example of this workflow for a simple case.

To begin, let's assume you have a Job Profile that is configured with a single MS Smooth output with Playready DRM enabled. When you retrieve its XML via the REST interface, you obtain an XML that looks like the example below:

```
<?xml version="1.0" encoding="UTF-8"?>
<job_profile version="2.0.2.xxxx" product="Elemental Server">
  <name>Basic MS Smooth</name>
  <permalink>basic_ms_smooth</permalink>
  <description>One MS Smooth output with DRM</description>
  <timecode_config>
    <source>embedded</source>
  </timecode_config>
  <priority>50</priority>
  <stream_assembly>
    <name>stream_assembly_0</name>
    <video_description>
      <afd_signaling>None</afd_signaling>
      <anti_alias>true</anti_alias>
      <drop_frame_timecode>true</drop_frame_timecode>
      <fixed_afd nil="true"/>
      <height>360</height>
      <insert_color_metadata>false</insert_color_metadata>
      <respond_to_afd>None</respond_to_afd>
      <selected_gpu nil="true"/>
      <stretch_to_output>false</stretch_to_output>
      <timecode_passthrough>false</timecode_passthrough>
      <vbi_passthrough>false</vbi_passthrough>
      <width>640</width>
      <h264_settings>
        <adaptive_quantization>high</adaptive_quantization>
        <bitrate>800000</bitrate>
        <buf_fill_pct nil="true"/>
        <buf_size nil="true"/>
        <cabac>true</cabac>
        <flicker_reduction>off</flicker_reduction>
        <force_field_pictures>false</force_field_pictures>
        <framerate_denominator>1</framerate_denominator>
        <framerate_follow_source>false</framerate_follow_source>
        <framerate_numerator>30</framerate_numerator>
        <gop_closed_cadence>1</gop_closed_cadence>
        <gop_num_b_frames>2</gop_num_b_frames>
        <gop_size>60</gop_size>
        <interpolate_frc>false</interpolate_frc>
        <look_ahead_rate_control>medium</look_ahead_rate_control>
        <max_bitrate nil="true"/>
        <max_qp nil="true"/>
        <min_qp nil="true"/>
```

```
      <num_ref_frames>1</num_ref_frames>
      <par_denominator>1</par_denominator>
      <par_follow_source>false</par_follow_source>
      <par_numerator>1</par_numerator>
      <passes>1</passes>
      <qp nil="true"/>
      <qp_step nil="true"/>
      <scd>true</scd>
      <slices>1</slices>
      <slow_pal>false</slow_pal>
      <telecine>None</telecine>
      <profile>Main</profile>
      <rate_control_mode>CBR</rate_control_mode>
      <gop_mode>fixed</gop_mode>
      <interlace_mode>progressive</interlace_mode>
    </h264_settings>
    <codec>h.264</codec>
    <video_preprocessors>
      <deinterlacer>
        <algorithm>interpolate</algorithm>
        <deinterlace_mode>Deinterlace</deinterlace_mode>
      </deinterlacer>
      <noise_reducer>
        <filter>Bilateral</filter>
        <strength>3</strength>
      </noise_reducer>
    </video_preprocessors>
  </video_description>
  <audio_description>
    <language_code nil="true"/>
    <order>1</order>
    <stream_name>audio_1</stream_name>
    <aac_settings>
      <bitrate>64000</bitrate>
      <channels>2</channels>
      <latm_loas>false</latm_loas>
      <mpeg2>false</mpeg2>
      <sample_rate>44100</sample_rate>
      <profile>LC</profile>
      <rate_control_mode>CBR</rate_control_mode>
    </aac_settings>
    <codec>aac</codec>
  </audio_description>
</stream_assembly>
<output_group>
  <name nil="true"/>
  <order>1</order>
  <ms_smooth_group_settings>
    <content_key>ee939e0d-52ff-4b04-b01a-22e2e51674c6</content_key>
    <custom_attributes/>
    <drm_system>playready</drm_system>
    <encryption_type>AES-128-CTR</encryption_type>
    <fragment_length>2</fragment_length>
    <initial_iv>1</initial_iv>
    <iv_size>64</iv_size>
    <key_id>ee939e0d-52ff-4b04-b01a-22e2e51674c6</key_id>
    <key_seed/>
    <keyprovider_type/>
    <license_url>http://my_license_server.com</license_url>
    <manifest_encoding>utf-8</manifest_encoding>
    <ui_license_url/>
```

```
      <destination>
        <uri>/data/server/outgoing/ms_smooth/</uri>
      </destination>
    </ms_smooth_group_settings>
    <type>ms_smooth_group_settings</type>
    <output>
      <description nil="true"/>
      <extension>ismv</extension>
      <log_edit_points>false</log_edit_points>
      <name_modifier>_ms_smooth_basic</name_modifier>
      <order>1</order>
      <output_subdirectory nil="true"/>
      <container>ismv</container>
      <stream_assembly_name>stream_assembly_0</stream_assembly_name>
    </output>
  </output_group>
</job_profile>
```

To transform this Job Profile XML into a valid Job XML several items must be modified. First, the root tag of the xml must be changed from job_profile to job. Next, the permalink and description tags should be removed. The name tag must also be removed. Finally, you must add at least one input to your Job.

Following the above basic steps, the XML is now valid to create a new Job. At this point you may also update any of the fields in the XML to suit the specific needs of your Event. In this example we will update the content_key, key_id, and publish_point URI fields within the MS Smooth Group settings, and the bitrate within the video codec settings. After we update the XML it should look like the following example. The few places that required modification in this case are highlighted:

```
<?xml version="1.0" encoding="UTF-8"?>
<job version="2.0.2.xxxx" product="Elemental Server">
  <input>
    <file_input>
      <uri>/data/server/incoming/my_video.mp4</uri>
    </file_input>
  </input>
  <timecode_config>
    <source>embedded</source>
  </timecode_config>
  <priority>50</priority>
  <stream_assembly>
    <name>stream_assembly_0</name>
    <video_description>
      <afd_signaling>None</afd_signaling>
      <anti_alias>true</anti_alias>
      <drop_frame_timecode>true</drop_frame_timecode>
      <fixed_afd nil="true"/>
      <height>360</height>
      <insert_color_metadata>false</insert_color_metadata>
      <respond_to_afd>None</respond_to_afd>
      <selected_gpu nil="true"/>
      <stretch_to_output>false</stretch_to_output>
      <timecode_passthrough>false</timecode_passthrough>
      <vbi_passthrough>false</vbi_passthrough>
      <width>640</width>
      <h264_settings>
        <adaptive_quantization>high</adaptive_quantization>
        <bitrate>900000</bitrate>
        <buf_fill_pct nil="true"/>
        <buf_size nil="true"/>
        <cabac>true</cabac>
        <flicker_reduction>off</flicker_reduction>
        <force_field_pictures>false</force_field_pictures>
        <framerate_denominator>1</framerate_denominator>
```

```
      <framerate_follow_source>false</framerate_follow_source>
      <framerate_numerator>30</framerate_numerator>
      <gop_closed_cadence>1</gop_closed_cadence>
      <gop_num_b_frames>2</gop_num_b_frames>
      <gop_size>60</gop_size>
      <interpolate_frc>false</interpolate_frc>
      <look_ahead_rate_control>medium</look_ahead_rate_control>
      <max_bitrate nil="true"/>
      <max_qp nil="true"/>
      <min_qp nil="true"/>
      <num_ref_frames>1</num_ref_frames>
      <par_denominator>1</par_denominator>
      <par_follow_source>false</par_follow_source>
      <par_numerator>1</par_numerator>
      <passes>1</passes>
      <qp nil="true"/>
      <qp_step nil="true"/>
      <scd>true</scd>
      <slices>1</slices>
      <slow_pal>false</slow_pal>
      <telecine>None</telecine>
      <profile>Main</profile>
      <rate_control_mode>CBR</rate_control_mode>
      <gop_mode>fixed</gop_mode>
      <interlace_mode>progressive</interlace_mode>
    </h264_settings>
    <codec>h.264</codec>
    <video_preprocessors>
      <deinterlacer>
        <algorithm>interpolate</algorithm>
        <deinterlace_mode>Deinterlace</deinterlace_mode>
      </deinterlacer>
      <noise_reducer>
        <filter>Bilateral</filter>
        <strength>3</strength>
      </noise_reducer>
    </video_preprocessors>
  </video_description>
  <audio_description>
    <language_code nil="true"/>
    <order>1</order>
    <stream_name>audio_1</stream_name>
    <aac_settings>
      <bitrate>64000</bitrate>
      <channels>2</channels>
      <latm_loas>false</latm_loas>
      <mpeg2>false</mpeg2>
      <sample_rate>44100</sample_rate>
      <profile>LC</profile>
      <rate_control_mode>CBR</rate_control_mode>
    </aac_settings>
    <codec>aac</codec>
  </audio_description>
</stream_assembly>
<output_group>
  <name nil="true"/>
  <order>1</order>
  <ms_smooth_group_settings>
    <content_key>ee939e0d-52ff-4b04-b01a-22e2e51674c7</content_key>
    <custom_attributes/>
    <drm_system>playready</drm_system>
```

```
      <encryption_type>AES-128-CTR</encryption_type>
      <fragment_length>2</fragment_length>
      <initial_iv>1</initial_iv>
      <iv_size>64</iv_size>
      <key_id>ee939e0d-52ff-4b04-b01a-22e2e51674c7</key_id>
      <key_seed/>
      <keyprovider_type/>
      <license_url>http://my_license_server.com</license_url>
      <manifest_encoding>utf-8</manifest_encoding>
      <ui_license_url/>
      <destination>
        <uri>/data/server/outgoing/ms_smooth/</uri>
      </destination>
    </ms_smooth_group_settings>
    <type>ms_smooth_group_settings</type>
    <output>
      <description nil="true"/>
      <extension>ismv</extension>
      <log_edit_points>false</log_edit_points>
      <name_modifier>_ms_smooth_basic</name_modifier>
      <order>1</order>
      <output_subdirectory nil="true"/>
      <container>ismv</container>
      <stream_assembly_name>stream_assembly_0</stream_assembly_name>
    </output>
  </output_group>
</job>
```

This modified XML can now be submitted via either REST or the web interface to create your new Job with your specific updated settings.

JOB PROFILES

Job profiles can be used for commonly used job settings. The permalink of a job profile may be substituted for its id.

URL	METHOD	PARAMETERS	RETURNS	DESCRIPTION
/job_profiles	GET	Pagination parameters	Job Profiles List	Retrieves a list of job profiles in the system
/job_profiles	POST	Job Profile Parameters	Job Profile Description	Creates a new job profile
/job_profiles/<id>	GET		Job Profile Description	Retrieves a specific job profile in the system
/job_profiles/<id>	PUT	Job Profile Parameters	Job Profile Description	Updates an existing job profile with new settings
/job_profiles/<id>	DELETE			Deletes job profile <id>

PRESETS

Presets define commonly used settings for outputs and stream assemblies. The permalink of a preset may be substituted for its id.

URL	METHOD	PARAMETERS	RETURNS	DESCRIPTION
/presets	GET	Pagination parameters	Preset List	Retrieves a list of presets in the system
/presets	POST	Preset Parameters	Preset Description	Creates a new preset
/presets/<id>	GET		Preset Description	Retrieves a specific preset in the system

URL	METHOD	PARAMETERS	RETURNS	DESCRIPTION
/presets/<id>	PUT	Preset Parameters	Preset Description	Updates an existing preset with new settings
/presets/<id>	DELETE			Deletes preset <id>

PRESET CATEGORIES

Preset categories allow for the sorting of presets.

URL	METHOD	PARAMETERS	RETURNS	DESCRIPTION
/preset_categories	GET	Pagination parameters	Preset Category List	Retrieves a list of presets categories, and displays the list of presets associated with each category.
/preset_categories	POST	Preset Category Parameters	Preset Category Description	Creates a new preset category
/preset_categories/<id>	GET		Preset Category Description	Retrieves a specific preset category and displays its list of presets
/preset_categories/<id>	PUT	Preset Category Parameters	Preset Category Description	Updates an existing preset category with new settings
/preset_categories/<id>	DELETE			Deletes preset category <id>

JOB WATCH FOLDERS

Job watch folders automatically apply a given job profile to any media files that are placed within a pre-defined directory.

URL	METHOD	PARAMETERS	RETURNS	DESCRIPTION
/job_watch_folders	GET	Pagination parameters	Job Watch Folder List	Retrieves a list of job watch folders in the system
/job_watch_folders	POST	Job Watch Folder Parameters	Job Watch Folder Description	Creates a new job watch folder
/job_watch_folders/<id>	GET		Job Watch Folder Description	Retrieves a specific job watch folder in the system
/job_watch_folders/<id>	PUT	Job Watch Folder Parameters	Job Watch Folder Description	Updates an existing job watch folder with new settings
/job_watch_folders/<id>	DELETE			Deletes job watch folder <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, Sequencer Settings, (Cluster Settings if part of a cluster)	Retrieves information about the current system settings. This XML is in a format that is accepted by the configure script (-i <filename>). This can be used to configure many identical boxes.

URL	METHOD	RETURNS	DESCRIPTION
/settings?cluster=true	GET	Timezone, Network Settings, Firewall Settings, Mount Point Settings, Authentication Settings, Sequencer Settings, Cluster Settings	Retrieves information about the current system settings including example cluster settings configured such that the current server is the master node. This can be used to help configure Slave nodes after the Master node has been configured.
/settings/network	GET	Network Settings	Retrieves information about the current network settings. Other Elemental Server units can communicate on the interface marked <management_interface> in a clustered environment.
/settings/mount_points	GET	Mount Point Settings	Retrieves information about the current mount point settings.
/settings/firewall	GET	Firewall Settings	Retrieves information about the current firewall settings.
/settings/snmp	GET	SNMP Settings	Retrieves information about the current SNMP settings.
/settings/authentication	GET	Authentication Settings	Retrieves information about the current authentication settings.
/settings/advanced	GET	Sequencer Settings	Retrieves information about the current sequencer settings.
/settings/cluster	GET	Cluster Settings	Retrieves information about the current cluster settings. Only available for clustered systems.
/settings/stop	POST	<stop></stop>	Enables a graceful shutdown of the Elemental service. Currently running jobs will run to completion, but no new jobs will be started. When all jobs have completed the service will shut down.
/settings/start	POST	<start></start>	Sends a start command to the Elemental service. Used to restart the service after a /settings/stop command.

ALERTS AND MESSAGES

The alerts API provides information about current alert conditions on the Elemental Server or nodes in a cluster. Messages provide an audit list of events on the Server or nodes in a cluster.

URL	METHOD	PARAMETERS	RETURNS	DESCRIPTION
/alerts	GET	Filter parameters can be appended to the URL, eg: /alerts?filter=all&node_filter=2	List of alerts	Active (or all if filter=all) alerts for the cluster (or node if node_filter set).
/messages	GET	Filter parameters can be appended to the URL, eg: /messages?filter=Error&node_filter=2	List of messages	Messages can be Errors, Warnings, or Audit messages. They have a code and a text message, and are associated with a particular node or job. See Codes for common messages codes.

ERROR CODES

The following list details common errors that the system may report. Error messages that contain *italics* in the following table are dynamic at runtime and will contain more details on the specific error.

CODE	ERROR MESSAGE	TROUBLESHOOTING
1010	Unable to open input file *filename*.	The Elemental Media Engine (EME) could not open the input file. Check the input URI and permissions to open it.
1020	No video stream in input file.	Elemental products require at least one video stream in the input file. Audio only streams are not supported.
1030	Unknown or unsupported video codec.	Check that this product supports the input source audio and video codec. See Input Codecs for a list of valid input media
1040	*Details on invalid setting*	One or more processing settings are not supported or compatible with the job. Details are given in the error message. If you see this error, please contact an Elemental support technician with details.
1050	*Disk full at SERVER_INSTALL_DIR*	The directory where the environment SERVER_INSTALL_DIR points to, must have at least 10MBytes of free space to start the EME.
1055	*Error sending output to dest.uri*	Check that the credentials are correct, the user has permissions to write to the destination URI and that the system supports the URI.
1060	*Input clipping region not found in input.*	The start and end timecodes specified in the Clip Input section of each Input must exist in the associated input stream.
1070	Adobe HDS configuration errors	Adobe HDS configuration parameters are incorrect. Details are given in the error message.
1999	*Critical EME (Elemental Media Engine) error*	This code is returned for errors that require an Elemental support technician to continue troubleshooting.
2010	Job *job.id* has too many outputs to run successfully. Please split the outputs into two or more jobs.	The job will require too many resources to run as one instance. It should be broken up into smaller jobs so the system can work on the job in pieces.
2030	Processing script *script_location* returned message: *message*	A pre or post processing script returned a message. This error comes from a custom pre or post script that has been executed before or after the job.
2040	Error deleting file from *input.uri* Check sequencer log for more details	There was an error deleting the input source file during a post processing step. Check that the elemental user has permissions to delete the file.
2050	Error copying file from *input.uri* to *dest.uri*: *more details...*	Check that the elemental user has permissions to move the file from the input to the destination URI and that the system supports the URI.
2080	Licensing Error	A licensing error/issue is raised when either the licenses doesn't support the hardware, the installed software, or a trial license has expired.
2999	*Critical sequencer error*	This code is returned for errors that require an Elemental support technician to continue troubleshooting.

WARNING CODES

The following list are common warnings that the system may report

CODE	WARNING MESSAGE	TROUBLESHOOTING
102010	Problem with pre-processing: *more details...*	There was a problem with the pre-processing script. The system will continue with the job, and more details can be found in the sequencer.output log file.
102020	Problem with post-processing: *more details...*	There was a problem with the post-processing script. The system will continue with the job, and more details can be found in the sequencer.output log file.
102030	Processing script *script_location* returned message: *message*	A pre or post processing script returned a message. This warning comes from a custom pre or post script that has been executed before or after the job.

CODE	WARNING MESSAGE	TROUBLESHOOTING
102040	This job is being updated with timestamps in the future relative to the management node.	The node where the job is running has a different system time than the management node. This can cause errors with managing stale jobs. It can be solved by ensuring all nodes in the cluster are set to the same time and/or are using the same NTP server.
102050	Input file does not yet exist on this server.	The input file could not be found to generate a preview image. Since the input may be there in the future because of a preprocessing script or other outside automation, this is only a warning. If the input is not available when the EME runs the job, then the system logs an error
102070	GPU selection overridden	User assigned GPU was overridden by the system. This can happen if the chosen GPU is disabled in advanced settings, or if the system combines image processing with another stream for efficiency.

QUERY PARAMETERS

The Elemental Server REST Interface allows for a series of query parameters to be appended to certain GET requests. These query parameters can be combined together for advanced querying.

PAGINATION

All 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:

```
/jobs?page=3&per_page=30
```

FILTER

Jobs can be filtered by state, for a more focused set of results. The parameters used to filter the results can be adjusted by appending the filter parameter to the end of the request:

```
/jobs?filter=active
```

At this time, only a single filter parameter is allowed per request. The set of valid filter values are listed below.

FILTER	DESCRIPTION
pending	Jobs in the pending state
active	Jobs in the preprocessing, running or postprocessing state
pre	Jobs in the preprocessing state
running	Jobs in the running state
post	Jobs in the postprocessing state
complete	Jobs in the complete state
cancelled	Jobs in the cancelled state
error	Jobs in the error state
archived	Jobs that have been archived

AUTHENTICATION AND REST

When authentication is enabled on the Elemental Server 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://<server>/api/v2.0/jobs/1?clean=true by the user 'admin' with the api_key '1acpJN7oEDn3BDDYhQ' that expires on June 1, 2011 UTC. In this case the url parameter is '/jobs/1' and the X-Auth-Expires value is '1306886400'. Thus the value of X-Auth-Key should be computed as follows:

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

Authenticated requests using the X-Auth headers are not persisted. If another request needs to be made, the the X-Auth-Key must be recalculated and all the headers must be set correctly again.

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 '/jobs/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://<server>/jobs/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://<server>/jobs
```

PARAMETERS

- Job
  - Location
  - Input
    Video Selector | Audio Selector | Audio Selector Group | Input Clipping
  - Timecode Config
  - Notification | Pre-Process | Post-Process | Image Inserter | Avail Blanking
  - Stream Assembly
    - Video Description
      - Rectangle
      - H.264 Settings | VC1 Settings | MPEG-2 Settings
        ProRes Settings | Frame Capture Settings | Uncompressed Settings
      - Video Preprocessor
        - Color Corrector | Image Inserter | Deinterlacer | Noise Reducer
          Watermarking | Timecode Burnin
    - Audio Description
      - AAC Settings | WAV Settings | AIFF Settings | WMA2 Settings
        MPEG-1 Layer II Settings | Dolby Digital Settings | Dolby Digital Plus Settings
        DTSE Settings | Pass Through Settings
      - Remix Settings | Audio Normalization Settings
    - Caption Description
      - Embedded Source Settings | File Source Settings | Teletext Source Settings
        DVB Sub Source Settings | Ancillary Source Settings
      - Burnin Destination Settings | Scc Destination Settings
  - Output Group
    - File Group Settings | Apple Live Group Settings | MS Smooth Group Settings
      HDS Group Settings
    - Flash Access Settings
    - Verimatrix Settings | Secure Media Settings | Irdeto Settings | Conax Settings
      Generic Keyprovider Settings | Self Generated Settings
    - Output | External Output
      - Apple Live Settings
      - MP4 Settings | F4V Settings | MOV Settings | UVU Settings
      - M2TS Settings | M3u8 Settings | TS Settings
- Preset | Preset Category | Remix Settings Preset
- Job Profile
- Job Watch Folder
- Format Identifier Parameters

PARAMETERS

The following tables outline parameters that can be set for objects in Elemental Server. 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. Default values are shown in **bold**.

JOB

NAME	TYPE	RANGE	DESCRIPTION
preroll_input	Input		Preroll media added before main input content.
input	Input		Job input parameters. There can be multiple <input> sections added to a job. These inputs will be concatenated together to create the output video. See URI Types for supported protocols.
postroll_input	Input		Postroll media added after main input is complete.
timecode_config	Timecode Config		Contains settings used to acquire and adjust timecode information from inputs.
profile	integer or string	Valid Profile ID, name, or permalink	If included, only an input parameter is needed. NOTE: Can only be used for creation. A valid ID, name or permalink must be provided. To ensure accuracy, always use permalinks that are distinct from profile names.
priority	integer	1 – 100 (Default: **50**)	Priority indicates the order that pending Jobs will be processed. 100 is highest priority.
user_data	string		User-defined data to be attached to the job. This data is available with job status requests.
notification	Notification		Settings for notification on status changes of this job.
pre_process	Pre-Process		Settings for preprocessing steps.
post_process	Post-Process		Settings for postprocessing steps.
image_inserter	Image Inserter		Settings for the image inserter. When attached to a Job, inserts images into the decoded input and appears in every output. Note that using multiple input files (or preroll/postroll files) with different resolutions will cause this image to be scaled differently depending on the input. Assigning image inserters to individual output files is recommended in this case.
avail_blanking	Avail Blanking		Settings for ad avail blanking. Video can be blanked or overlayed with an image, and audio muted during SCTE-35 triggered ad avails.
server_output	Output		Settings for a single output. One job may contain many server output sections.
avsync_enable	boolean	**true** or false	Enables A/V sync.
avsync_pad_trim_audio	boolean	**true** or false	Pads or trims audio to match the video duration.
stream_assembly	Stream Assembly		A Stream assembly for this job. A job can have several stream assemblies which define output codec settings.
output_group	Output Group		An output group for this Job. Output groups contain information about where streams should be distributed.

LOCATION

```
<job>
  <destination>
    <uri>/data/server/outgoing/</uri>
  </destination>
  ...
</job>
```

NAME	TYPE	RANGE	DESCRIPTION
uri	string		Uniform Resource Identifier (e.g. /data/server/input.mp4) This should be a path to a file accessible to the Elemental Server system either on the local filesystem or through a SMB mount.
username	string		Username if credentials are required to access file.
password	string		Password if credentials are required to access file.

URI TYPES

For content source and destination locations, Elemental Server can support file transfers using the protocols in the following table. Network protocols (other than Streaming Inputs as described below) will cache the file to the local filesystem as a preprocessing or postprocessing step.

STREAMING INPUTS

HTTP Live Streaming inputs (eg, http://<web server>[:port]/path/file.m3u8[?bitrate=20000&retries=10&retry_interval=2]) will be progressively downloaded while transcoding. If the file pointed to by the URI is a variant playlist, the highest bitrate stream will be chosen as the source; the optional bitrate argument can be used to select a specific stream from the playlist. The retries and retry_interval parameters control how many times the transcode will retry downloading on HTTP errors, and how many seconds it will wait between retries.

PROTOCOL	URI FORMAT	INPUT/ DESTINATION	AUTHENTICATION
Local file	/data/server/folder/file.ext	Both	None
CIFS or NFS mounted filesystem	/mnt/mountpoint/file.ext	Both	See Mount Point Settings
HTTP	http://<web server>[:port]/path/file.ext	Input only	Basic
HTTPS	https://<web server>[:port]/path/file.ext	Input only	Basic
FTP	ftp://<ftp server>[:port]/path/file.ext	Both	Basic
SFTP	sftp://<ftp server>[:port]/path/file.ext	Both	SSH authentication; host key authentication (add /home/elemental/.ssh/id_rsa.pub to sftp server's authorized key list)
SCP	scp://<ftp server>[:port]/path/file.ext	Both	SSH authentication; host key authentication (add /home/elemental/.ssh/id_rsa.pub to sftp server's authorized key list)
Amazon S3	s3://<bucket>/<object>	Both	Enter the Access Key ID in the username field. Enter the Secret Access Key in the password field.

INPUT

NAME	TYPE	RANGE	DESCRIPTION
file_input	Location		File to transcode. Must be accessible to the Elemental Server node that is running the job, either on the local disk or as a SMB mount. See URI Types for supported protocols.
order	integer		Specifies the order which this input is placed when concatenating multiple input files. Not valid for preroll and postroll inputs.
program_id	integer		Selects a specific program from a multi-program transport stream. If the program doesn't exist, will default to first program within the transport stream.
deblock_enable	boolean	true or **false**	Turns on the deblocking filter for this input. MPEG-2 inputs have the deblocking filter enabled by default, unless specifically disabled in the Advanced Settings page.
deblock_strength	integer	-5 – **5**	Adjusts the magnitude of deblocking from -5 to 5, with 0 being the nominal value.
input_clipping	Input Clipping		Specifies additional clipping information.
video_selector	Video Selector		Specifies a particular video stream within an input source. An input may have only a single video selector.
audio_selector	Audio Selector		Specifies a particular audio stream within an input source. An input may have multiple audio selectors.
audio_selector_group	Audio Selector Group		Specifies set of audio selectors within an input to combine. An input may have multiple audio selector groups. See Audio Selector Group for more information.

VIDEO SELECTOR

A video selector allows for fine-grained control of exactly what video data is extracted from an input.

NAME	TYPE	RANGE	DESCRIPTION
program_id	integer		Selects a specific program from within a multi-program transport stream. If the program doesn't exist, the first program within the transport stream will be selected by default.
color_space	string	**follow**, rec_601, rec_709	Specifies the colorspace of an input. This setting works in tandem with Color Corrector > color_space_conversion to determine if any conversion will be performed.

AUDIO SELECTOR

An audio selector allows for fine-grained control of exactly what audio data is extracted from an input.

NAME	TYPE	RANGE	DESCRIPTION
order	integer	> 0	Required when an input has multiple audio selectors. The order is important when merging audio sources using an Audio Selector Group since it determines the order of channels in the resulting output.

NAME	TYPE	RANGE	DESCRIPTION
default_selection	boolean	true or **false**	When an Audio Description specifies an audio source and no matching AudioSelector or AudioSelectorGroup is found in the input, then the audio selector marked as “default” will be used. If none are marked as default, silence will be inserted for the duration of the input.
pid	decimal integer	> 0	Selects a specific PID from within an audio source (e.g. 257 selects PID 0×101).
track	string	Comma separated string of integers > 0 (Default: first English track or first track if none are marked English).	Desired track or tracks to process as shown by the media info. If multiple tracks are listed, they will be combined (e.g. combining two mono tracks into a stereo track).
offset	integer	integer	Specifies a time delta in milliseconds to offset the audio from the input video.
external_audio_file_input	Location		Specifies audio data from an external file source.

AUDIO SELECTOR GROUP

An audio selector group is used to specify a set of audio data sources within an input that will be combined. Each audio selector group *must* be given a name, and every audio selector within a group *must* share the same offset value. Multiple audio selectors can be included in a group by specifying multiple audio_selector_names. A group’s combined audio can then be used in any Audio Description by specifying the group name in the audio_source_name.

NAME	TYPE	RANGE	DESCRIPTION
name	string	non-empty string	A name for the grouping of audio selectors. The name is used when specifying an audio source in an Audio Description.
audio_selector_name	string	non-empty string	Name of an Audio Selector within the same input to include in the group. Audio selector names are standardized, based on their order within the input (e.g. “Audio Selector 1”). The audio_selector_name parameter can be repeated to add any number of audio selectors to the group.

INPUT CLIPPING

NAME	TYPE	RANGE	DESCRIPTION
start_timecode	string	valid timecode	Specifies the timecode at which video processing should begin. Should be specified as either embedded, zero-based, or specifiedstart, which is determined by the timecode_source value for the Job. The timecode must be of the format NN:NN:NN:NN with values *hour*:*minute*:*second*:*frame*. For specifiedstart, the start value is attached to the first input frame. Either start_timecode or end_timecode may be left blank, but not both.
end_timecode	string	valid timecode	Specifies the timecode at which video processing should end. Should be specified as either embedded, zero-based, or specifiedstart, which is determined by the timecode_source value for the Job. The timecode must be of the format NN:NN:NN:NN with values *hour*:*minute*:*second*:*frame*. Either start_timecode or end_timecode may be left blank, but not both.

TIMECODE CONFIG

NAME	TYPE	RANGE	DESCRIPTION
source	string	**embedded**, zerobased, specifiedstart	Determines method of acquiring timecode information.
start	string	valid timecode	Determines starting timecode when source has value specifiedstart. The timecode must be of the format NN:NN:NN:NN with values *hour*:*minute*:*second*:*frame*. If an anchor value is present, then the start is used in conjunction with the anchor to calculate an initial timecode for the output. If no anchor value is present, then the start is used as the initial timecode for the output. Starting timecode is also used for input clipping.
anchor	string	valid timecode	Determines timecode of frame used for anchoring. That frame (on input) will have the same timecode on output, even if rate conversion is in effect. If source is specifiedstart, then that is assumed to the timecode of the first input frame. If source is zerobased, then the timecode of the first input frame will be assumed to be 00:00:00:00. If source is embedded, then the timecode value on the first input frame will be used.

NOTIFICATION

Notification objects allow Elemental Server to notify a user via email or an automated workflow system by HTTP POST of the status of a Job.

NAME	TYPE	RANGE	DESCRIPTION
email	string	A list of valid email addresses, comma separated	Email address(es) to send notifications.
web_callback_url	string	A valid HTTP URL	URL to call for notifications. Job status XML will be POSTed to this address when the selected events occur.
on_started	boolean	true or **false**	Send notification when Job starts.
on_complete	boolean	true or **false**	Send notification when Job completes.
on_error	boolean	true or **false**	Send notification when Job encounters an error.
on_warning	boolean	true or **false**	Send notification when Job encounters a warning.
on_cancel	boolean	true or **false**	Send notification when Job is cancelled.

PRE-PROCESS

NAME	TYPE	RANGE	DESCRIPTION
copy_local	boolean	true or **false**	Copies input file to the local disk before transcoding. Useful if input source is on a network drive. FTP inputs are copied to the local disk by default.
script	Location		Script to run before Job starts.

POST-PROCESS

NAME	TYPE	RANGE	DESCRIPTION
delete_source	boolean	true or **false**	Deletes input source file(s) after the Job is complete. Note this will not affect preroll or postroll source files.
processed	Location		Location to move input source file to after Job completes. Note this will not affect preroll or postroll source files.
script	Location		Script to run after the Job completes.

AVAIL BLANKING

NAME	TYPE	RANGE	DESCRIPTION
enabled	boolean	true or **false**	Indicates video, audio and captions will be blanked during SCTE-35 triggered ad avails.
avail_blanking_image	Location		Blanking image to be used. Leave empty for solid black. Only bmp and png images are supported.

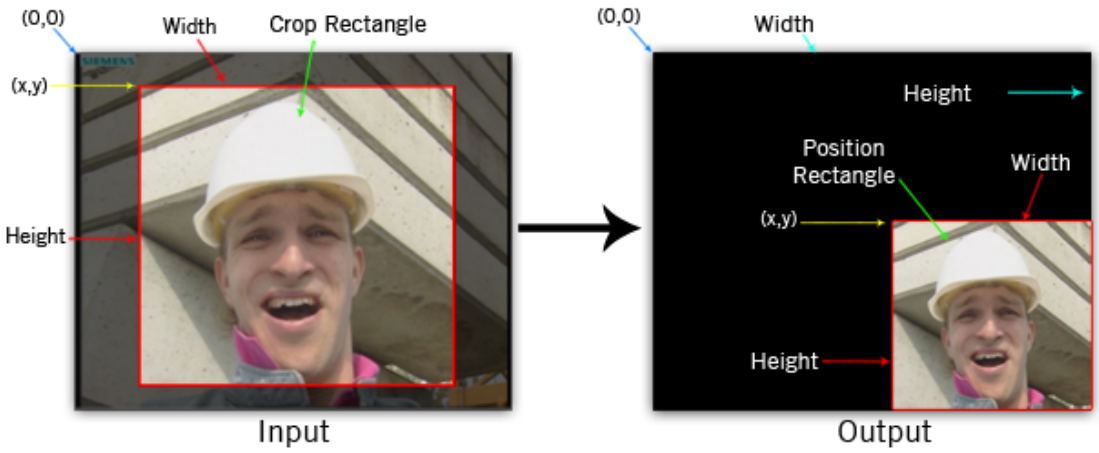
STREAM ASSEMBLY

A stream assembly describes the audio and video settings for an output stream

NAME	TYPE	RANGE	DESCRIPTION
name	string		Stream name. This is used to attach streams to outputs. This field is not saved, it is replaced with an id field once the Job is saved. See stream_assembly_name under Output.
video_description	Video Description		Video settings for this stream.
audio_description	Audio Description		Audio settings for this stream. There can be multiple audio settings in a single stream.
caption_description	Caption Description		Caption settings for this stream. There can be multiple caption settings in a single stream.
preset	integer or string	A valid Preset ID or name	Preset values to use for this stream. If this is included, no further parameters are needed. If parameters are included, they will override the Preset value. A valid ID or name must be provided, specifying by permalink is not supported.

VIDEO DESCRIPTION

Video description contains the settings needed for a video stream in an output media. The following picture shows how crop, position and width and height relate to each other. If crop or position is not given, the software will ensure the display aspect ratio is preserved in the resolution specified by height and width.



NAME	TYPE	RANGE	DESCRIPTION
codec	enum	h.264, mpeg2, vc1, frame capture, uncompressed, prores	Video codec. See Video Codecs for supported output codecs.
***codec*_settings**	Codec Settings	h264_settings, vc1_settings, mpeg2_settings, frame_capture_settings, uncompressed_settings	Codec specific settings. Note: replace *codec* with the codec you are using in the XML tag (e.g. <h264_settings>).
width	integer	32 – 4096 (Default: source video width)	Output video width (in pixels). Leave blank to use source video width. Display aspect ratio is always preserved by letterboxing or pillarboxing when necessary.
height	integer	32 – 3112 (Default: source video height)	Output video height (in pixels). Leave blank to use source video height.
stretch_to_output	boolean	true or **false**	Automatically configures the output position Rectangle to stretch the video to the specified output resolution. This prevents the output from being letterboxed or pillerboxed. This option will override any position value.
anti_alias	boolean	true or **false**	Use the anti-aliasing scaler. This should be used with large downscaling ratios.
vbi_passthrough	boolean	true or **false**	Passes user data fields from input source to output source. This includes 608 & 708 closed caption data. Framerate must be set to Follow Source or must be 50 fps or greater.
timecode_passthrough	boolean	true or **false**	Passes timecode data from input source to output source. Timecode will be adjusted for framerate conversion, however there will be no guarantee that the source timecode is frame-accurate with the output.
drop_frame_timecode	boolean	**true** or false	Instructs timecode insertion to use drop-frame timecodes for 29.97 fps outputs. If it is not possilbe to use drop-frame timecodes, the system will fall back on non-drop-frame and note the discrepancy in the logs.
crop	Rectangle		Crop input to rectangle. Aspect ratio preservation is disabled when this parameter is used.

NAME	TYPE	RANGE	DESCRIPTION
position	Rectangle		Position output in rectangle. Aspect ratio preservation is disabled when this parameter is used.
video_preprocessors	Video Preprocessors		Video preprocessing to apply to this output.
cpu_encode	boolean	true or **false**	Indicates that this stream should be encoded using the CPU rather than the GPU.
respond_to_afd	string	**None**, Respond, Passthrough	Indicates how to respond to the AFD values in the input stream. Respond causes input video to be clipped, depending on AFD value, input display aspect ratio and output display aspect ratio.
afd_signaling	string	**None**, Auto, Fixed	Indicates that AFD values will be written into the output stream. In the case where respond_to_afd is Auto, the system will try to preserve the input AFD value (in cases where multiple AFD values are valid). Only valid for H.264 and MPEG2 outputs.
fixed_afd	integer	0 – 15	Four bit AFD value to write on all frames of video in the output stream. Only valid when afd_signaling is set to 'Fixed'.
insert_color_metadata	boolean	true or **false**	Includes colorspace metadata in the output.

RECTANGLE

NAME	TYPE	RANGE	DESCRIPTION
x	integer		Left of rectangle.
y	integer		Top of rectangle.
width	integer		Width of rectangle.
height	integer		Height of rectangle.

H.264 SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
profile	enum	Baseline, **Main**, High	H.264 Profile.
level	enum	**Auto**, 1, 1.1, 1.2, 1.3, 2, 2.1, 2.2, 3, 3.1, 3.2, 4, 4.1, 4.2, 5, 5.1, 5.2	H.264 Level.
rate_control_mode	enum	VBR, **CBR**, CQ, ABR	Rate control mode. CQ uses constant quantizer (qp), ABR (average bitrate) does not write HRD parameters.
bitrate	integer	>= 1000 (Default: **5000000**)	Average bitrate in bits/second. Required for VBR, CBR, and ABR.
max_bitrate	integer		Maximum bitrate in bits/second (for VBR mode only).
buf_size	integer		Size of buffer (HRD buffer model).
buf_fill_pct	integer	0 – 100	Percentage of the buffer that should initially be filled (HRD buffer model).
framerate_numerator	integer		Framerate numerator – framerate is a fraction, e.g. 24000 / 1001 = 23.976 fps.
framerate_denominator	integer		Framerate denominator.
framerate_follow_source	boolean	**true** or false	No framerate conversion from source.

NAME	TYPE	RANGE	DESCRIPTION
interpolate_frc	boolean	true or **false**	Interpolates during a framerate conversion. Produces smoother motion during a framerate change.
telecine	string	**None**, Soft, or Hard	Use to encode using soft or hard telecine (Hard telecine produces 29.97i output from 23.976 input. Soft telecine output is 23.976 and conversion to 29.97i is done by the player).
slow_pal	boolean	true or **false**	Enables Slow PAL rate conversion. 23.976 input is relabeled as 25 and audio is sped up correspondingly.
interlace_mode	enum	**progressive**, top_field, bottom_field, follow_top_field, follow_bottom_field	Interlace mode. top_field and bottom_field force top or bottom field dominance, follow_top_field and follow_bottom_field will use the source video's field dominance if source is interlaced, otherwise will force top or bottom.
gop_mode	enum	**Fixed** or Follow	Fixed uses the specified gop_size, Follow matches the I-frame interval of source. Follow is not recommended for intra-only sources
gop_size	integer	>= 0 (Default: **80**)	GOP Length (keyframe interval, in frames).
gop_num_b_frames	integer	0 – 7 (Default: **2**)	Number of B-frames between reference frames.
gop_closed_cadence	integer	>= 0 (Default: **1**)	Frequency of closed GOPs. In streaming applications, it is recommended that this be set to 1 so a decoder joining mid-stream will receive an IDR frame as quickly as possible. Setting this value to 0 will break output segmenting.
min_i_interval	integer	0 – 30 (Default: **0**)	Min I-interval. Allows insertion of scene change I-frame with the specified I-interval else results in GOP shrink/stretch. Note: Maximum GOP stretch = GOP size + Min-I-interval – 1.
adaptive_quantization	string	**off**, low, medium, high	Adaptive quantization. Allows intra-frame quantizers to vary to improve visual quality.
flicker_reduction	string	**off**, low, high	Flicker reduction. Helps reduce I-frame pop. When enabled, recommended setting is **high**.
cabac	boolean	**true** or false	Enable CABAC (must be in Main or High profile).
qp	integer	1 – 51	Quantization parameter – fixed for CQ rate control mode, or starting QP for rate controller.
max_qp	integer	1 – 51	Maximum QP for rate controller.
min_qp	integer	1 – 51	Minimum QP for rate controller.
qp_step	integer	>= 1	Maximum QP change between frames for rate controller.
par_follow_source	boolean	**true** or false	No pixel aspect ratio conversion from source.
par_numerator	integer		Pixel Aspect Ratio numerator.
par_denominator	integer		Pixel Aspect Ratio denominator.
slices	integer	**1** – 32	Number of slices per picture. Must be less than or equal to the number of macroblock rows for progressive pictures, and less than or equal to half the number of macroblock rows for interlaced pictures.
scd	boolean	**true** or false	Scene change detection (inserts I-frames on scene changes).
look_ahead_rate_control	string	low, **medium**, high	Amount of lookahead. A value of low can decrease latency and memory usage, while high can produce better quality for certain content.
num_ref_frames	integer	**1** – 6	Number of reference frames to use. The encoder may use more than requested if using B-frames and/or interlaced encoding.
force_field_pictures	boolean	true or **false**	Disables PAFF/MBAFF encoding for interlaced outputs.
passes	integer	**1** or 2	Number of encoding passes.

VC1 SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
profile	enum	Simple, Main, **Advanced**	VC1 Profile.
level	enum	**Auto**, Low, Medium, High, L0, L1, L2, L3, L4	VC1 Level – For Advanced profile only Auto, L0, L1, L2, L3, L4 level selections are accepted. For Main profile only Auto, Low, Medium and High level selections are accepted. For Simple profile only Auto, Low and Medium selections are accepted.
rate_control_mode	enum	VBR, **CBR**, CQ, ABR	Rate control mode. CQ uses constant quantizer (qp), ABR (average bitrate) does not write HRD parameters.
bitrate	integer	>= 1000 (Default: **5000000**)	Average bitrate in bits/second. Required for VBR, CBR, and ABR.
max_bitrate	integer		Maximum bitrate in bits/second (for VBR mode only).
buf_size	integer		Size of buffer (HRD buffer model).
buf_fill_pct	integer	0 – 100	Percentage of the buffer that should initially be filled (HRD buffer model).
framerate_numerator	integer		Framerate numerator – framerate is a fraction, e.g. 24000 / 1001 = 23.976 fps.
framerate_denominator	integer		Framerate denominator.
framerate_follow_source	boolean	**true** or false	No framerate conversion from source.
interpolate_frc	boolean	true or **false**	Interpolates during a framerate conversion. Produces smoother motion during a framerate change.
slow_pal	boolean	true or **false**	Enables Slow PAL rate conversion. 23.976 input is relabeled as 25 and audio is sped up correspondingly.
interlace_mode	enum	**progressive**	Only progressive frames supported for VC-1.
gop_mode	enum	**Fixed** or Follow	Fixed uses the specified gop_size, Follow matches the I-frame interval of source. Follow is not recommended for intra-only sources.
gop_size	integer	>= 0 (Default: **80**)	GOP Length (keyframe interval, in frames).
gop_num_b_frames	integer	0 – 7 (Default: **2**)	Number of B-frames between reference frames.
gop_closed_cadence	integer	>= 0 (Default: **1**)	Frequency of closed GOPs. In streaming applications, it is recommended that this be set to 1 so a decoder joining mid-stream will receive an IDR frame as quickly as possible. Setting this value to 0 will break output segmenting.
min_i_interval	integer	0 – 30 (Default: **0**)	Min I-interval. Allows insertion of scene change I-frame with the specified I-interval else results in GOP shrink/stretch. Note: Maximum GOP stretch = GOP size + Min-I-interval – 1.
adaptive_quantization	string	**off**, low, medium, high	Adaptive quantization. Allows intra-frame quantizers to vary to improve visual quality.
qp	integer	1 – 31	Quantization parameter – fixed for CQ rate control mode, or starting QP for rate controller.
max_qp	integer	1 – 31	Maximum QP for rate controller.
min_qp	integer	1 – 31	Minimum QP for rate controller.
qp_step	integer	1 – 10	Maximum QP change between frames for rate controller.
par_numerator	integer		Pixel Aspect Ratio numerator.
par_denominator	integer		Pixel Aspect Ratio denominator.
par_follow_source	boolean	**true** or false	No pixel aspect ratio conversion from source.
scd	boolean	**true** or false	Scene change detection (inserts I-frames on scene changes).

NAME	TYPE	RANGE	DESCRIPTION
look_ahead_rate_control	string	low, **medium**, high	Amount of lookahead. A value of low can decrease latency and memory usage, while high can produce better quality for certain content.
passes	integer	**1** or 2	Number of encoding passes.

MPEG-2 SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
profile	enum	**Main**, 4:2:2	MPEG-2 Profile.
level	enum	**Auto**, Low, Main, High1440, High	MPEG-2 Level.
rate_control_mode	enum	VBR, **CBR**, CQ, ABR	Rate control mode. CQ uses constant quantizer (qp), ABR (average bitrate) does not write HRD parameters.
bitrate	integer	>= 1000 (Default: **5000000**)	Average bitrate in bits/second. Required for VBR, CBR, and ABR.
max_bitrate	integer		Maximum bitrate in bits/second (for VBR mode only).
buf_size	integer		Size of buffer (HRD buffer model).
buf_fill_pct	integer	0 – 100	Percentage of the buffer that should initially be filled (HRD buffer model).
framerate_numerator	integer		Framerate numerator – framerate is a fraction, e.g. 24000 / 1001 = 23.976 fps.
framerate_denominator	integer		Framerate denominator.
framerate_follow_source	boolean	**true** or false	No framerate conversion from source.
interpolate_frc	boolean	true or **false**	Interpolates during a framerate conversion. Produces smoother motion during a framerate change.
telecine	string	**None**, Soft, or Hard	Use to encode using soft or hard telecine (Hard telecine produces 29.97i output from 23.976 input. Soft telecine output is 23.976 and conversion to 29.97i is done by the player).
slow_pal	boolean	true or **false**	Enables Slow PAL rate conversion. 23.976 input is relabeled as 25 and audio is sped up correspondingly.
interlace_mode	enum	**progressive**, top_field, bottom_field, follow_top_field, follow_bottom_field	Interlace mode. top_field and bottom_field force top or bottom field dominance, follow_top_field and follow_bottom_field will use the source video's field dominance if source is interlaced, otherwise will force top or bottom.
gop_mode	enum	**Fixed** or Follow	Fixed uses the specified gop_size, Follow matches the I-frame interval of source. Follow is not recommended for intra-only sources.
gop_size	integer	>= 0 (Default: **12**)	GOP Length (keyframe interval, in frames).
gop_num_b_frames	integer	0 – 7 (Default: **2**)	Number of B-frames between reference frames.
gop_closed_cadence	integer	>= 0 (Default: **1**)	Frequency of closed GOPs. In streaming applications, it is recommended that this be set to 1 so a decoder joining mid-stream will receive an IDR frame as quickly as possible. Setting this value to 0 will break output segmenting.
min_i_interval	integer	0 – 30 (Default: **0**)	Min I-interval. Allows insertion of scene change I-frame with the specified I-interval else results in GOP shrink/stretch. Note: Maximum GOP stretch = GOP size + Min-I-interval – 1.
adaptive_quantization	string	**off**, low, medium, high	Adaptive quantization. Allows intra-frame quantizers to vary to improve visual quality.

NAME	TYPE	RANGE	DESCRIPTION
framing_quantization	float	**0.0** – 2.0	Framing quantization. Strength applied varies qp of the picture.
softness	integer	16 – 128	Softness. Generates quantiser matrix for intra, non-intra, luma and chroma blocks.
qp	integer	1 – 112	Quantization parameter – fixed for CQ rate control mode, or starting QP for rate controller.
max_qp	integer	1 – 112	Maximum QP for rate controller.
min_qp	integer	1 – 112	Minimum QP for rate controller.
qp_step	integer	1 – 10	Maximum QP change between frames for rate controller.
par_numerator	integer		Pixel Aspect Ratio numerator.
par_denominator	integer		Pixel Aspect Ratio denominator.
par_follow_source	boolean	**true** or false	No pixel aspect ratio conversion from source.
scd	boolean	**true** or false	Scene change detection (inserts I-frames on scene changes).
look_ahead_rate_control	string	low, **medium**, high	Amount of lookahead. A value of low can decrease latency and memory usage, while high can produce better quality for certain content.
d10_syntax	boolean	true or **false**	Produces a Type D-10 compatible bitstream (SMPTE 356M-2001).
passes	integer	**1** or 2	Number of encoding passes.

PRORES SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
framerate	ratio		Choose an output framerate in fps, or specify with a numerator and denominator. If “Follow Source” is selected, no framerate conversion from the source is performed.
par	ratio		Choose a predefined Pixel Aspect Ratio (PAR), or specify with a numerator and denominator. If “Follow Source” is selected, no PAR conversion from the source input is performed.
interpolate_frc	boolean	true or **false**	Interpolates during a framerate conversion. Produces smoother motion during a framerate change.
interlace_mode	enum	**progressive**, top_field, bottom_field, follow_top_field, follow_bottom_field	Interlace mode. top_field and bottom_field force top or bottom field dominance, follow_top_field and follow_bottom_field will use the source video’s field dominance if source is interlaced, otherwise will force top or bottom.
profile	enum	**Apple ProRes 422**, Apple ProRes 422 (HQ), Apple ProRes 422 (LT), Apple ProRes 422 (Proxy)	Apple ProRes Profile.
telecine	string	**None** or Hard	Use to encode using hard telecine (Hard telecine produces 29.97i output from 23.976 input.).
slow_pal	boolean	true or **false**	Enables Slow PAL rate conversion. 23.976 input is relabeled as 25 and audio is sped up correspondingly.

FRAME CAPTURE SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
quality	integer	1 – 100 (Default: **80**)	JPEG Quality – a higher value equals higher quality.

NAME	TYPE	RANGE	DESCRIPTION
instruction	string	Default: **“at 5s”**	Instructions in the form “(every|at) (number)(s|%|f)”, such as “every 5s” and “at 10%”. Units are ‘s’ for seconds, ‘%’ for percent, and ‘f’ for frame number. Impossible requests such as “every 0f” and “at 105%” are errors. Instructions can be combined with ‘or’, such as ‘at 10s or at 5s’. If the first instruction can not be satisfied (for example, the clip is 7s long), then the second instruction will be used. Instructions can *not* be combined with ‘and’. Create multiple Frame Capture outputs instead.
append_sequence_number	boolean		Appends a sequence number to frame capture files. Unchecking this box will overwrite the output file, which can be used to monitor transcode progress.

UNCOMPRESSED SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
framerate_numerator	integer		Framerate numerator – framerate is a fraction, e.g. 24000 / 1001 = 23.976 fps.
framerate_denominator	integer		Framerate denominator.
framerate_follow_source	boolean	**true** or false	No framerate conversion from source.
interpolate_frc	boolean	true or **false**	Interpolates during a framerate conversion. Produces smoother motion during a framerate change.
slow_pal	boolean	true or **false**	Enables Slow PAL rate conversion. 23.976 input is relabeled as 25 and audio is sped up correspondingly.

VIDEO PREPROCESSORS

NAME	TYPE	RANGE	DESCRIPTION
color_corrector	Color Corrector		Settings for the color corrector.
image_inserter	Image Inserter		Settings for the image inserter. When added here, applies to only this video stream.
deinterlacer	Deinterlacer		Settings for deinterlacer.
noise_reducer	Noise Reducer		Settings for noise_reducer.
watermarking	Watermarking		Embeds a unique and indelible digital watermark in the output.

COLOR CORRECTOR

NAME	TYPE	RANGE	DESCRIPTION
brightness	integer	1 – 100 (Default: **50**)	Brightness level.
contrast	integer	1 – 100 (Default: **50**)	Contrast level.
hue	integer	-180 – 180 (Default: **0**)	Hue in degrees.
saturation	integer	1 – 100 (Default: **50**)	Saturation level.
full_swing	boolean	true or **false**	*True* expands the input colorspace to *full-swing* encoding, *False* allows the input encoding to pass through.

NAME	TYPE	RANGE	DESCRIPTION
color_space_conversion	string	**none**, force_601, force_709	Determines if colorspace conversion will be performed. If set to *none*, no conversion will be performed. If *force_601* or *force_709* are selected, conversion will be performed for inputs with differing colorspaces. An input's colorspace can be specified explicitly in the Video Selector if necessary.

IMAGE INSERTER

The image inserter overlays a 32-bit Windows BMP or PNG file on the output video. The resolution of the image to be inserted must be smaller than the output resolution. When using Photoshop to output 32 bit .bmp files, be sure to set it to output the alpha channel. That's what keeps the logo from appearing inside a black or white box. An example image can be found in */opt/elemental_se/web/public/example_files/Elemental_logo.bmp*. An XML control file can also be used.

NAME	TYPE	RANGE	DESCRIPTION
image_inserter_input	Location		Image to insert. Must be 32 bit windows BMP, PNG, or XML control file.
image_x	integer		Placement of image on the horizontal axis in pixels. 0 is the left edge of the frame. Required for BMP and PNG input.
image_y	integer		Placement of image on the vertical axis in pixels. 0 is the top edge of the frame. Required for BMP and PNG input.
opacity	integer	0 – 100 (Default: **50**)	Opacity of image. 0 is transparent. 100 is fully opaque. Required for BMP and PNG input.

IMAGE INSERTER CONTROL FILE

When the image inserter input is an XML file, if can contain multiple commands to insert and remove images at specific times from the beginning of the stream. The following example inserts a logo 10 seconds into the stream, then removes it after 30 seconds:

```
<command value="image_inserter">
  <image_inserter>
    <index>0</index>
    <time>00:00:10.000</time>
    <image_inserter_input><uri>/opt/elemental_se/web/public/images/Elemental_logo1.bmp</uri></image_inse
    <activate>true</activate>
    <fade_ms>100</fade_ms>
    <opacity>100</opacity>
    <image_x>100</image_x>
    <image_y>100</image_y>
  </image_inserter>
  <image_inserter>
    <index>0</index>
    <time>00:00:40.000</time>
    <activate>false</activate>
    <fade_ms>100</fade_ms>
  </image_inserter>
</command>
```

NAME	TYPE	RANGE	DESCRIPTION
index	integer	0 or 1	The index of the image to enable or disable.
time	time	hh:mm:ss.fff	The time from the beginning of output stream the image should be enabled or disabled.
activate	boolean	true or false	Indicates whether this image is to be inserted (true) or removed (false).

NAME	TYPE	RANGE	DESCRIPTION
fade_ms	integer		The number of milliseconds across which the image will be faded in (if active is true) or out (if active is false).
image_inserter_input	Location		Image to insert. Must be 32-bit RGBA bitmap or PNG file (any bit depth). Only required when sending an insert command.
image_x	integer		Placement of image on the horizontal axis in pixels. 0 is the left edge of the frame. Only required when sending an insert command.
image_y	integer		Placement of image on the vertical axis in pixels. 0 is the top edge of the frame. Only required when sending an insert command.
opacity	integer	0 – 100	Opacity of image. 0 is transparent. 100 is fully opaque. Only required when sending an insert command.

DEINTERLACER

NAME	TYPE	RANGE	DESCRIPTION
deinterlace_mode	string	**Deinterlace**, Inverse Telecine, Adaptive	Deinterlace converts interlaced field pictures into progressive frame pictures. Inverse telecine removes telecine, converting from 29.97i video to 23.976p film. Adaptive detects if content is telecined or interlaced, and converts to progressive accordingly.
algorithm	enum	**interpolate**, blend, low_latency	Deinterlace algorithm (has no effect if deinterlace_mode is Inverse Telecine). Motion adaptive interpolate produces sharper pictures, while blend produces smoother motion. Low-latency is a linear interpolation over a single picture.

NOISE REDUCER

NAME	TYPE	RANGE	DESCRIPTION
filter	string	**Bilateral**, Mean, Gaussian, Lanczos, Conserve	Bilateral, Mean (or Box), Gaussian, Lanczos, conservative filters perform various spacial noise filtering functions.
strength	integer	**1** – 3	Relative strength of filtering (higher produces less noise, but blurrier images).

WATERMARKING

Digital watermarking embeds a unique and indelible identifier within a video that is recognizable by software but imperceptible to the eye. Content providers can use watermarks to track their media after it is distributed.

NAME	TYPE	RANGE	DESCRIPTION
provider	string	Civolution	Specifies a 3rd party watermarking provider. Currently, only Civolution is supported.
payload	integer		The unique watermarking integer identifier to embed in the video.
strength	integer	1 – 5	Specifies the strength of the watermarking algorithm. Stronger watermarking increases the chance of visible artifacts, but makes the watermark more resilient to re-encoding.

TIMECODE BURN-IN

NAME	TYPE	RANGE	DESCRIPTION
prefix	string	ASCII string	Specifies the prefix before the burned-in timecode. Prefixes accept ASCII characters from 0×20 to 0×7e (inclusive). The prefix will be inserted directly before the timecode. For example, a prefix of “EZ-” will result in the following timecode, “EZ-00:00:00:00”).
font_size	integer	**10**, 16, 32, 48	Determines the font size in pixels of the burned-in timecode.
position	string	**top_center**, top_left, top_right, middle_left, middle_center, middle_right, bottom_left, bottom_center, bottom_right	Determines position of the burned-in timecode relative to the output.

AUDIO DESCRIPTION

NAME	TYPE	RANGE	DESCRIPTION
codec	enum	aac, mp2, wma2, wav, aiff, ac3, ec3, pass through, dtse	Audio codec. See Audio Codecs for supported output codecs.
***codec*_settings**	Codec Settings	aac_settings, wav_settings, aiff_settings, wma2_settings, pass_through_settings, mp2_settings, ac3_settings, eac3_settings	Codec specific settings. Note: replace *codec* with the codec you are using in the XML tag (e.g. <aac_settings>).
order	integer	> 0	Required for multiple audio. Specifies the order the audio descriptions should be listed in.
language_code	string	ISO 639-2 three-digit code	Indicates the language of the audio output track.
stream_name	string	Alphanumeric characters, spaces, and underscore	Only used for MS Smooth outputs. Indicates the name displayed by the player (eg. English, or Director Commentary).
remix_settings	Remix Settings		Advanced audio remixing settings.
audio_source_name	string		Specifies which audio data to use from each input. In the simplest case, specify an Audio Selector by name based on its order within each input. For example if you specify “Audio Selector 3”, then the third audio selector will be used from each input. If an input does not have an “Audio Selector 3”, then the audio selector marked as “default” in that input will be used. If there is no audio selector marked as “default”, silence will be inserted for the duration of that input. Alternatively, an Audio Selector Group name may be specified, with similar default/ silence behavior. If no audio_source_name is specified, then “Audio Selector 1” will be chosen automatically.

NAME	TYPE	RANGE	DESCRIPTION
audio_normalization_settings	Audio Normalization Settings		Advanced audio normalization settings.

AAC SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
bitrate	integer	6000 – 1024000 (Default: **64000**)	Average bitrate in bits/second. Valid values depend on rate control mode and profile.
sample_rate	integer	8000 – 96000 (Default: **48000**)	Sample rate in hz. Valid values depend on rate control mode and profile.
channels	integer	1, **2**, 6	Mono, Stereo, or 5.1 channel layout. Valid values depend on rate control mode and profile.
rate_control_mode	enum	**CBR**, VBR	Rate Control Mode.
profile	enum	**LC**, HEV1, HEV2	AAC Profile.
latm_loas	boolean	true or **false**	Enables LATM / LOAS AAC output for raw or MPEG-2 Transport Stream (CableLabs) containers.
mpeg2	boolean	true or **false**	Use MPEG-2 AAC audio instead of MPEG-4 AAC audio for raw or MPEG-2 Transport Stream containers.
vbr_quality	enum	LOW1, LOW2, LOW3, MEDIUM1, MEDIUM2, MEDIUM3, HIGH1, HIGH2, HIGH3	VBR Quality Level – Only used if rate_control_mode is VBR.

WAV SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
sample_rate	integer	8000 – 192000 (Default: **44100**)	Sample rate in hz.
channels	integer	1, **2**	Mono or Stereo.
bit_depth	integer	**16** or 24	Bits per sample.

AIFF SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
sample_rate	integer	8000 – 192000 (Default: **44100**)	Sample rate in hz.
channels	integer	1, **2**	Mono or Stereo.
bit_depth	integer	**16** or 24	Bits per sample.

WMA2 SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
bitrate	integer	24000 – 1024000 (Default: **96000**)	Average bitrate in bits/second.

NAME	TYPE	RANGE	DESCRIPTION
sample_rate	integer	8000 – 48000 (Default: **44100**)	Sample rate in hz.
channels	integer	1, **2**	Mono or Stereo.

MPEG-1 LAYER II SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
bitrate	integer	32000 – 384000 (Default: **192000**)	Average bitrate in bits/second.
sample_rate	integer	32000 – **48000**	Sample rate in hz.
channels	integer	1, **2**	Mono or Stereo.

DOLBY DIGITAL AUDIO SETTINGS

Requires license

NAME	TYPE	RANGE	DESCRIPTION
bitrate	integer	64k – 640k (Default: **192k**)	Average bitrate in bits/second. Valid bitrates depend on the coding mode. Defaults for each coding mode are – 1_0: 96k, 1_1: 192k, 2_0: 192k, 3_2_LFE: 384k.
sample_rate	integer	Sample rate is always **48000**	Sample rate in hz.
coding_mode	string	1_0, 1_1, **2_0**, 3_2_LFE	Dolby Digital coding mode. Determines number of channels.
dynamic_range_compression	boolean	**true** or false	Adds dynamic range compression signaling to the output bitstream as defined in the Dolby Digital specification.
lfe_filter	boolean	true or **false**	Applies a 120Hz lowpass filter to the LFE channel prior to encoding. Only valid in 3_2_LFE mode.
dialnorm	integer	1 – 31	Sets the dialnorm for the output. If blank and input audio is Dolby Digital, dialnorm will be passed through.

DOLBY DIGITAL PLUS AUDIO SETTINGS

Requires license

NAME	TYPE	RANGE	DESCRIPTION
bitrate	integer	32k – 3024k (Default: **256k**)	Average bitrate in bits/second. Valid bitrates depend on the coding mode. Defaults for each coding mode are – 1_0: 128, 2_0: 128k, 3_2: 256k.
sample_rate	integer	Sample rate is always **48000**	Sample rate in hz.
coding_mode	string	1_0, 2_0, **3_2**	Dolby Digital Plus coding mode. Determines number of channels.
lfe_filter	boolean	**true** or false	Applies a 120Hz lowpass filter to the LFE channel prior to encoding. Only valid with 3_2 coding mode.
dialnorm	integer	1 – 31	Sets the dialnorm for the output. If blank and input audio is Dolby Digital Plus, dialnorm will be passed through.
dc_filter	boolean	**true** or false	Activates a DC highpass filter for all input channels.

NAME	TYPE	RANGE	DESCRIPTION
bandwidth_filter	boolean	**true** or false	Activates a lowpass filter with a cutoff near the specified audio bandwidth that is applied to the main input channels.
drc_line	string	none, **film_standard**, film_light, music_standard, music_light, speech	Enables Dynamic Range Compression that restricts the absolute peak level for a signal.
drc_rf	string	none, **film_standard**, film_light, music_standard, music_light, speech	Enables Heavy Dynamic Range Compression, ensures that the instantaneous signal peaks do not exceed specified levels.
surround_mode	string	**not_indicated**, enabled, disabled	When encoding 2/0 audio, sets whether Dolby Surround is matrix encoded into the two channels.
lfe	boolean	**true** or false	When encoding 3/2 audio, enables the LFE channel
surround_ex_mode	string	not_indicated, enabled, **disabled**	When encoding 3/2 audio, sets whether an extra center back surround channel is matrix encoded into the left and right surround channels.
stereo_downmix	string	**not_indicated**, lo_ro, lt_rt, dpl2	Stereo downmix preference. Only used for 3/2 coding mode.
lt_rt_center_mix_level	float	3.0, 1.5, 0.0, -1.5, **-3.0**, -4.5, -6.0, -60	Left total/Right total center mix level. Only used for 3/2 coding mode.
lt_rt_surround_mix_level	float	-1.5, **-3.0**, -4.5, -6.0, -60	Left total/Right total surround mix level. Only used for 3/2 coding mode.
lo_ro_center_mix_level	float	3.0, 1.5, 0.0, -1.5, **-3.0**, -4.5, -6.0, -60	Left only/Right only center mix level. Only used for 3/2 coding mode.
lo_ro_surround_mix_level	float	-1.5, **-3.0**, -4.5, -6.0, -60	Left only/Right only surround mix level. Only used for 3/2 coding mode.
phase_shift_90_degree	boolean	**true** or false	Applies a 90-degree phase shift to the surround channels. Only used for 3/2 coding mode.
attenuate_3_db	boolean	true or **false**	Applies a 3 dB attenuation to the surround channels. Only used for 3/2 coding mode.

DTS EXPRESS SETTINGS

Requires license

NAME	TYPE	RANGE	DESCRIPTION
bitrate	integer	48000 – 768000 (Default: **192000**)	Average bitrate in bits/second
sample_rate	integer	44100, **48000**	Sample rate in hz. Only 48000 is supported in Ultraviolet containers.
bit_depth	integer	**16** or 24	Bits per sample.
channel_layout	string	C, **L_R**, L_R_C_LFE_Ls_Rs	DTS channel layout. Determines number of channels.
dynamic_range_compression	boolean	**true** or false	Adds dynamic range compression signaling to the output bitstream as defined in the DTS specification.
dialnorm	integer	1 – 31	Sets the dialnorm for the output. If blank and input audio is DTS Express, dialnorm will be passed through.

PASS THROUGH SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
name	string		Pass through settings require a name and no other parameters; this is a known issue that will be addressed in a future release.

REMIX SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
channels_in	integer	1 to **16**	Number of input channels to be used.
channels_out	integer	1, **2**, 6	Number of output channels to be produced.
channel_mapping	XML		Remixing values to use. Units are in dB and acceptable values are within the range from -60 (mute) and 6 dB. See example.
preset	integer		Remix Settings Preset ID. If this is included, do not include any other parameters.

The channel mapping parameter takes a variable XML structure that represents the array of input channels to output channels. Units are in dB and acceptable values are within the range from -60 (mute) and 6 dB. An example for default stereo is displayed below:

```
<channel_mapping>
  <out_ch_0>
    <in_ch_0>0</in_ch_0>
    <in_ch_1>-60</in_ch_1>
  </out_ch_0>
  <out_ch_1>
    <in_ch_0>-60</in_ch_0>
    <in_ch_1>0</in_ch_1>
  </out_ch_1>
  ...
</channel_mapping>
```

AUDIO NORMALIZATION SETTINGS

Requires license

NAME	TYPE	RANGE	DESCRIPTION
algorithm	string	**1770-1** or 1770-2	Audio normalization algorithm to use. 1770-1 conforms to the CALM Act specification, 1770-2 conforms to the EBU R-128 specification.
correct_audio	boolean	**true** or false	When enabled the output audio is corrected using the chosen algorithm. If disabled, the audio will be measured but not adjusted.
target_lkfs	float	-59 to 0	Target LKFS to adjust volume to. If no value is entered, a default value will be used according to the chosen algorithm. The CALM Act (1770-1) recommends a target of -24 LKFS. The EBU R-128 specification (1770-2) recommends a target of -23 LKFS.
real_time_correction	boolean	true or **false**	If true, the audio is measured and corrected simultaneously in one pass. If false, an analysis pass is run first to determine the Integrated LKFS for the entire audio stream before it is corrected during a second pass.

NAME	TYPE	RANGE	DESCRIPTION
log_loudness	boolean	**true** or false	Log each output's audio track loudness to a CSV file.
truepeak	boolean	true or **false**	Calculate and log the TruePeak for each output's audio track loudness.

CAPTION DESCRIPTION

NAME	TYPE	RANGE	DESCRIPTION
order	integer	> 0	Required for multiple captions. Specifies the order the caption descriptions should be listed in.
source_type	string	**Embedded**, SCC, Teletext, DVB-Sub, Ancillary, TTML, STL, SRT, SMI	Indicates the source to pull captions from. Supported values are 'Embedded', which pulls EIA-608/708 captions embedded in the elementary video stream, 'SCC', 'STL', 'SRT', 'SMI' and 'TTML' which pull from an external file, 'DVB-Sub', and 'Teletext' which attaches to transport stream (Archive and UDP) outputs.
***source*_settings**	Source Settings	embedded_source_settings, file_source_settings, teletext_source_settings, dvb_sub_source_settings, ancillary_source_settings	Specific settings required by the specific source type. Note: replace *source* with the source type you are using in the XML tag. If using SCC source_type, then use the file_source_settings.
***destination*_settings**	Destination Settings	burnin_destination_settings	Specific settings required by the Burn-In destination type.
language_code	string	ISO 639-2 three-digit code	Indicates the language of the caption output track.
language_description	string	Alphanumeric characters, spaces, and underscore	Only used for MS Smooth outputs. Indicates the name displayed by the player (eg. English, or Spanish).

MS SMOOTH TTML TEMPLATES

There are two template files used by Elemental Server to generate TTML. They contain the style information applied to subtitles that the player will then render. The two files are:

- `/opt/elemental_se/config/template-ttml-head.txt`
- `/opt/elemental_se/config/template-ttml-foot.txt`

The two files in isolation can be considered text files. One is the header, which contains the actual CSS definitions. The other is the footer, which contains just a few closing tags and which generally would never need to be modified. When concatenated, they should produce well-formatted XML. The specific caption text XML of a video will be injected in between the two template files.

Your template XML must be well-formed, otherwise the system will fall back to an internal XML template. No additional validation beyond well-formedness is performed.

EMBEDDED SOURCE SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
destination_type	string	**Embedded**, Embedded+SCTE-20, SRT, SMI, SMPTE-TT, TTML, CFF-TT, WebVTT, SCC, Burn-In	Destination format for captions. 'Embedded' and 'Embedded+SCTE-20' indicate that the captions will be inserted into the elementary stream, 'Burn-In' indicates the captions will be burned into the output, other values indicate that external files will be generated. Captions with an external file destination must be specified using a separate caption-only output. TTML outputs are SMTPE-TT compliant. CFF-TT outputs are only compatible with the Ultraviolet container.
source_608_track_number	integer	**1**	Specifies the video track index used for extracting captions. The system only supports one input video track, so this should always be set to '1'.
source_608_channel_number	integer	**1** – 4	Specifies the 608/708 channel number within the video track from which to extract captions.

FILE SOURCE SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
destination_type	string	**Embedded**, Embedded+SCTE-20, SRT, SMI, SMPTE-TT, TTML, CFF-TT, WebVTT, SCC, Burn-In	Destination format for captions. 'Embedded' and 'Embedded+SCTE-20' indicate that the captions will be inserted into the elementary stream, 'Burn-In' indicates the captions will be burned into the output, other values indicate that external files will be generated. Captions with an external file destination must be specified using a separate caption-only output.
source_file	Location		External caption file used for loading captions. Accepted file extentions are 'scc', 'ttml', 'dfxp', 'stl', 'srt', and 'smi'.
time_delta	integer		Specifies a time delta in seconds to offset the captions from the source file.

TELETEXT SOURCE SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
destination_type	string	SRT, SMI, SMPTE-TT, TTML, CFF-TT, WebVTT, SCC, Burn-In, **Teletext**	Destination format for captions. 'Teletext' indicates teletext passthrough, 'Burn-In' indicates the captions will be burned into the output, other values indicate that external files will be generated. Captions with an external file destination must be specified using a separate caption-only output.
page_number		100 to 8FF	Specifies the teletext page number within the data stream from which to extract captions. Must be a three-digit hexadecimal string.

DVB SUB SOURCE SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
destination_type	string	**DVB-Sub**, Burn-In, CFF-TT	A value of DVB-Sub indicates DVB-Sub passthrough.

ANCILLARY SOURCE SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
destination_type	string	**Embedded**, Embedded+SCTE-20, SRT, SMI, SMPTE-TT, TTML, CFF-TT, WebVTT, SCC, Burn-In	Destination format for captions. 'Embedded' and 'Embedded+SCTE-20' indicate that the captions will be inserted into the elementary stream, 'Burn-In' indicates the captions will be burned into the output, other values indicate that external files will be generated. Captions with an external file destination must be specified using a separate caption-only output.

BURN-IN DESTINATION SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
font	Location		External font file used for caption burn-in. File extension must be 'ttf' or 'tte'.
font_size	string	**auto**, or a positive integer	When set to **auto**, font_size will scale depending on the size of the output. Giving a positive integer will specify the exact font size in points.
alignment	string	**centered**, left	If no explicit x_position or y_position is provided, setting alignment to centered will place the captions at the bottom center of the output. Similarly, setting a left alignment will align captions to the bottom left of the output. If x and y positions are given in conjunction with the alignment parameter, the font will be justified (either left or centered) relative to those coordinates.
x_position	integer	integer value greater than or equal to 0	Specifies the horizontal position of the caption relative to the left side of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the left of the output. If no explicit x_position is provided, the horizontal caption position will be determined by the alignment parameter.
y_position	integer	integer value greater than or equal to 0	Specifies the vertical position of the caption relative to the top of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the top of the output. If no explicit y_position is provided, the caption will be positioned towards the bottom of the output.
font_color	string	**white**, black, yellow, red, green, blue, black	Specifies the color of the burned-in captions.
font_opacity	integer	0 – **255**	Specifies the opacity of the burned-in captions. 255 is opaque; 0 is transparent.
background_color	string	**none**, black, white	Specifies the color of the rectangle behind the captions.
background_opacity	integer	0 – 255	Specifies the opacity of the background rectangle. 255 is opaque; 0 is transparent. Leaving this parameter blank is equivalent to setting it to 0 (transparent).
shadow_color	string	**none**, black, white	Specifies the color of the shadow cast by the captions.
shadow_opacity	integer	0 – 255	Specifies the opacity of the shadow. 255 is opaque; 0 is transparent. Leaving this parameter blank is equivalent to setting it to 0 (transparent).
shadow_x_offset	integer	integer value	Specifies the horizontal offset of the shadow relative to the captions in pixels. A value of -2 would result in a shadow offset 2 pixels to the left.

NAME	TYPE	RANGE	DESCRIPTION
shadow_y_offset	integer	integer value	Specifies the vertical offset of the shadow relative to the captions in pixels. A value of -2 would result in a shadow offset 2 pixels below the text.

SCC DESTINATION SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
framerate	string	24, 29.97 dropframe, 29.97 non-dropframe	Timecode framerate of output SCC file

OUTPUT GROUP

NAME	TYPE	RANGE	DESCRIPTION
name	string		Output group name. This is used to attach media to output groups. This field is not saved, it is replaced with an id field once the job is saved. See output_group_name under Output.
*output_group*_settings	Group Settings	file_group_settings, apple_live_group_settings, hds_group_settings, ms_smooth_group_settings	Output group type-specific settings. Note: replace *output_group* with the group type you are using in the XML tag.

FILE GROUP SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
destination	Location		A directory and base filename where archive files should be written. Destination URI fields accept Format Identifiers. If the base filename portion of the URI is left blank, the base filename of the first input will be automatically inserted.
rollover_interval	integer	> 1	Number of seconds to write to archive file before closing and starting a new one. Leave blank to disable archive rollover.

APPLE LIVE GROUP SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
destination	Location		A directory and base filename where archive files should be written. Destination URI fields accept Format Identifiers. If the base filename portion of the URI is left blank, the base filename of the first input will be automatically inserted.
segment_length	integer	>= 1 (Default: **10**)	Length of MPEG-2 Transport Stream segments to create (in seconds). Note that segments will end on the next keyframe after this number of seconds, so actual segment length may be longer.
base_url	string		A partial URI prefix that will be prepended to each output in the base .m3u8 file. Can be used if streams are delivered from a different URL than the main .m3u8 file.

NAME	TYPE	RANGE	DESCRIPTION
floating_point_manifest	boolean	**true** or false	Indicates whether the output manifest should use floating point values for segment duration.
use_subdirectories	boolean	true or **false**	Place segments in subdirectories.
segments_per_subdirectory	integer	>= 1	Number of segments to write to a subdirectory before starting a new one. use_subdirectories must be true for this setting to have an effect.
insert_program_date_time	boolean	true or **false**	Inserts EXT-X-PROGRAM-DATE-TIME tag in .m3u8 manifest files. The program date time value is derived from the input timecode source.
encryption_type	string	AES-128, SAMPLE-AES	Encrypts the segments with the given encryption scheme. Leave blank to disable. Selecting 'Disabled' in the web interface also disables encryption.
key_rotation_count	integer	> 0 (Default: **3**)	For use with encryption_type. The AES encryption key will rotate after this many segments. Set to 0 to use the same key throughout the entire encoding session. This parameter will be ignored when the key provider is Conax or Irdeto.
iv_follows_segment_number	boolean	**true** or false	For use with encryption_type. 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		For use with encryption_type. This is a 128-bit, 16-byte hex value representented by a 32-character text string. If iv_follows_segment_number is set to false then this parameter is required and is used as the IV for encryption.
key_provider_settings	Key Provider Settings	self_generated_settings, verimatrix_settings, secure_media_settings, irdeto_settings, conax_settings, generic_keyprovider_settings	Key Provider-specific settings.
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 delmited list of version values (1/2/3).
key_save_location	Location		The location where key files will be saved. Value is accepted only when no key provider (self-generated) is specified.
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. Value is accepted only when no key provider (self-generated) is specified.
ad_markers	string	adobe, elemental	Choose one or more ad marker types to pass SCTE35 signals through to this group of Apple HLS outputs.

MICROSOFT SMOOTH STREAMING GROUP SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
destination	Location		A directory and base filename where archive files should be written. Destination URI fields accept Format Identifiers. If the base filename portion of the URI is left blank, the base filename of the first input will be automatically inserted.
fragment_length	integer	>= 1 (Default: **2**)	Length of mp4 fragments to generate (in seconds). Fragment length must be compatible with GOP size and framerate.
drm_system	string	**nil** or playready	A value of playready enables Microsoft Playready DRM. Playready requires key_id and either key_seed or content_key.
encryption_type	string	**nil** or AES-128-CTR	Encrypts the fragments with the given encryption scheme when using Microsoft Playready DRM. Only used when drm_system is set to playready, and when playready is enabled the default is AES-128-CTR.
iv_size	integer	**64, 128**	Number of bits to use in the IV.
initial_iv	integer	Default: **1**	Initial value of IV.
key_id	string	UUID/GUID	Specifies a key ID to use for Playready DRM, must be a valid UUID/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.
key_provider_settings	Key Provider Settings	irdeto_settings, conax_settings	Key Provider-specific settings.
manifest_encoding	string	**utf-8** or utf-16	Text encoding to use for server and client manifests

HDS GROUP SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
destination	Location		A directory and base filename where archive files should be written. Destination URI fields accept Format Identifiers. If the base filename portion of the URI is left blank, the base filename of the first input will be automatically inserted.
fragment_length	integer	>= 1 (Default: **2**)	Length of fragments to generate (in seconds). Fragment length must be compatible with GOP size and Framerate.
segment_length	integer	>= 1 (Default: **30**)	Length of f4f segments to create (in seconds). Note that segments will end on the next keyframe after this number of seconds, so actual segment length may be longer. Setting this value to 0 will place all fragments in a single f4f file.
encryption_type	string	**nil** or hds_encryption	Encrypts the fragments with the given encryption scheme when using Adobe HDS DRM.
encryption_settings	Encryption Settings	flash_access_settings	Encryption-specific settings.

NAME	TYPE	RANGE	DESCRIPTION
stream_level_manifest	string	1.0,2.0,3.0	Version to specify which player will be used. Set to version 1.0 for interoperation with most existing players and version 3.0 for interoperation with Primetime players.

FLASH ACCESS SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
video_encrypt_level	string	**low**, medium, high	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	string		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.
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	Location		Credentials for the Adobe HDS license server.
packager_credential	Location		Credentials for the Adobe HDS packager.
transport_certificate	Location		The transport certificate, in DER format.
policy_file	Location		A file which contains the rules and restrictions that determine how, when, and where protected content can be viewed by consumers.
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.

VERIMATRIX SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
resourceid	string		Verimatrix Resource ID.
verimatrix_server	Location		The Verimatrix server that will provide the keys.
reuse_last_key	boolean	**true** or false	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	DESCRIPTION
resourceid	integer	0 – 4294967295	Secure Media Resource ID.
secure_media_server	Location		The Secure Media server that will provide the keys.
reuse_last_key	boolean	**true** or false	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	DESCRIPTION
service_url	Location	URL with login credentials	Specifies the Location of the Irdeto server. Both a URL and login credentials are required.
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*	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	true or **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.

CONAX SETTINGS

NAME	TYPE	RANGE	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	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** or false	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.

SELF-GENERATED SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
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.
key_save_location	Location		The location where key files will be saved.

OUTPUT

An output object describes the settings for a single output file in a job.

NAME	TYPE	RANGE	DESCRIPTION
name_modifier	string		String concatenated to destination filename. Only required if the job has more than one output. Accepts Format Identifiers.
stream_assembly_name	string		Name of the stream to attach to this output. This field is not saved, it is replaced with an id field once the Job is saved. See name field in Stream Assembly.
extension	string		Output file extension. If empty, this will be auto-selected from the container type.
description	string		Description of this output.
container	enum	mp4, f4v, wmv, raw, m2ts, m3u8, ismv, ts, mov, uvu	Container for this output. See Containers for supported output containers. Can be auto-detected from extension field. Certain containers require a *container*_settings object. If not specified, the default object will be created.
*container*_settings	Container Settings	mp4_settings, f4v_settings, mov_settings, uvu_settings, m2ts_settings, ts_settings	Container specific settings. Note: replace *container* with the container you are using in the XML tag (e.g. <mp4_settings>).
output_settings	Output Settings	apple_live_settings	Specific settings for this type of output. Can only be used if this server output contains an Apple HLS group.
scte35_passthrough	boolean	true or **false**	If true, passes any SCTE-35 signals from the input source to this output. Only available for certain containers.
klv_passthrough	boolean	true or **false**	If true, passes any KLV data from the input source to this output. Only available for certain containers.
ebif_passthrough	boolean	true or **false**	If true, passes any EBIF data from the input source to this output. Only available for certain containers.
nielsen_id3_passthrough	boolean	true or **false**	If true, Nielsen inaudible tones for media tracking will be detected in the input audio and an equivalent ID3 tag will be inserted in the output. Only available for certain containers.
log_edit_points	boolean	true or **false**	Generates an XML file in the job log directory with initial timecode, timecode of input switches, and final timecode. This can be used to for later editing of this output.
order	integer	> 0	Sets the order of output creation from lowest to highest when a job has multiple outputs.

EXTERNAL OUTPUTS

Allows an output not being produced in this output group to be added to the variant playlist. Can be used to generate an .m3u8 playlist with backup streams from an external encoder, or to share streams between multiple output groups.

NAME	TYPE	RANGE	DESCRIPTION
order	integer	> 0	Required for multiple outputs within an output group. Specifies the order in which the output should be listed within the output group. Outputs and external outputs are ordered together.
external uri	string		URI for external output entry in .m3u8 playlist. Can be fully-qualified or relative.
bandwidth	string		Bandwidth value for external output in .m3u8 playlist.

APPLE LIVE SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
audio_only_image	Location		For use with an audio only Stream. Must be a .jpg or .png file. If given, this image will be used as the cover-art for the audio only output. Ideally, it should be formatted for an iPhone screen for two reasons. The iPhone does not resize the image, it crops a centered image on the top/bottom and left/right. Additionally, this image file gets saved bit-for-bit into every 10-second segment file, so will increase bandwidth by {image file size} * {segment count} * {user count.}.
alternate_audio_track	boolean	true or **false**	Treat an audio only apple live output as an alternate audio track.

MP4 SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
progressive_downloading	boolean	**true** or false	If true, ensures blocks are written in the order required for progressive downloading.

F4V SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
progressive_downloading	boolean	true or **false**	If true, ensures blocks are written in the order required for progressive downloading.

MOV SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
reference	string	**self_contained** or external	A value of ‘external’ creates separate media files and the wrapper file (.mov) contains references to these media files. A value of ‘self_contained’ creates only a wrapper (.mov) file and this file contains all of the media.
include_clap	boolean	true or **false**	Include ‘clap’ atom if appropriate for the video output settings.

UVU SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
metadata	string		XML formatted metadata. An external file of metadata information can also be supplied by specifying the file location inside of a file node (e.g. <file>/data/server/path/to/metadata.xml</file>).
apid	string		Application Identifier
base_location	string		Base Location, limited to 256 characters
purchase_location	string		Purchase Location, limited to 256 characters
encypted	boolean	true or **false**	Indicates whether the Ultraviolet outputs should use CFF Common Encryption.
iv_size	integer	64, **128**	Number of bits to use in the IV.

NAME	TYPE	RANGE	DESCRIPTION
video_content_key	hexadecimal string		The content key to use for video encryption. If left blank, a content key will be autogenerated.
video_key_id	hexadecimal string		The key ID to use for video encryption. If left blank, a key ID will be autogenerated.
use_same_keys	boolean		Use the same content key and key id for both audio and video.
audio_content_key	hexadecimal string		The content key to use for audio encryption. If left blank, a content key will be autogenerated.
audio_key_id	hexadecimal string		The key ID to use for audio encryption. If left blank, a key ID will be autogenerated.
content_key	hexadecimal string		If specified, this content key will be used for both video and audio encryption
key_id	hexadecimal string		If specified, this key ID will be used for both video and audio encryption.
frame_height	integer	32 – 3112 (Default: source video height)	Output frame height (in pixels). Leave blank to use source video height.
frame_width	integer	32 – 4096 (Default: source video width)	Output frame width (in pixels). Leave blank to use source video width.
fragment_length	float	1 – 3.003	Length of fragments to generate (in seconds). Fragment length must be compatible with GOP size and framerate.
kdf_template	Location		Location of an external Key Description File. If not specified, the default KDF template located at /opt/elemental_se/config/template-kdf.xml will be used.

UVU SETTINGS METADATA FORMAT

The metadata field accepts an XML string. The accepted fields are documented below.

The following fields are automatically generated:

```
ContentID           : Content identifier
APID                : Application identifier
Publisher           : Publisher
ReleaseYear         : Year of mediaâ€™s release (Required)
TitleDisplay19      : Title, limited to 19 characters
TitleDisplay60      : Title, limited to 60 characters
TitleSortable       : Sortable title keywords separated by commas
Summary190          : Description of media, limited to 190 characters
DescriptionLanguage : Language code used for this metadata
```

Additional settings available, but are not required are:

```
ReleaseDate      : Month, day, year and time of release (time is optional)
Summary400       : Description of media, limited to 400 characters
CopyrightLine    : Displayable copyright line
WorkType         : Type of work
SequenceInfo     : Sequence information (required for some work types)
Parent           : Metadata for parent items (required for some work types)
AltIdentifier    : Other identifiers for this content
RatingSet        : Content rating set
PictureColorType : Picture color type
PictureFormat    : Picture format
BaseLocation     : Base location, limited to 256 characters
PurchaseLocation : Purchase location, limited to 256 characters
Chapter[@index]  : Chapter marker at this many seconds
                   (index is a zero-based attribute counting each chapter)
```

M2TS SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
bitrate	integer	Default: **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.
pmt_pid	integer	0 – 8190 (Default: **480**)	Packet Identifier (PID) for the Program Map Table (PMT) in the transport stream.
video_pid	integer	0 – 8190 (Default: **481**)	Packet Identifier (PID) of the elementary video stream in the transport stream.
audio_pid	integer	0 – 8190 (Default: **482**)	Packet Identifier (PID) of the elementary audio stream in the transport stream.
scte35_pid	integer	0 – 8190	Packet Identifier (PID) of the SCTE-35 stream in the transport stream.
timed_metadata_pid	integer	0 – 8190	Packet Identifier (PID) of the timed metadata stream in the transport stream.
private_metadata_pid	integer	0 – 8190	Packet Identifier (PID) of the private metadata stream in the transport stream.
program_num	integer	0 – 65535	The value of the program number field in the Program Map Table.
pcr_every_pes	boolean	**true** or false	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.
pcr_pid	integer	0 – 8191	Packet Identifier (PID) of the Program Clock Reference (PCR) in the transport stream.
transport_stream_id	integer	0 – 65535	The value of the transport stream ID field in the Program Map Table.
psi_repeat_rate	float	0 – 1000 (Default: **8.0**)	The number of Program Specific Information tables (PSIs) inserted in the transport stream each second. Note a value of 0 will insert at the beginning of the stream only.
vbr	boolean	true or **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	true or **false**	When true, uses DVB buffer model for Dolby Digital audio. When false, the ATSC model is used.
null_packet_bitrate	float	>= **0**	Value in bits per second of extra null packets to insert into the transport stream. This can be used if a downstream encryption system requires periodic null packets.
audio_packets_per_pes	integer	>= 0 (Default: **2**)	The number of audio packets to insert for each PES packet.
segmentation_time	float	> 0	The length in seconds of each segment. Required unless markers is set to *none*.
fragment_time	float	>= 0	The length in seconds of each fragment. Only used with EBP markers.
segmentation_markers	string	none, rai_segstart, rai_adapt, psi_segstart, or ebp	The strategy for inserting segmentation markers.
es_rate	boolean	true or **false**	Include the ES Rate field in the PES header.

M3U8 SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
pmt_pid	integer	0 – 8190 (Default: **480**)	Packet Identifier (PID) for the Program Map Table (PMT) in the transport stream.
video_pid	integer	0 – 8190 (Default: **481**)	Packet Identifier (PID) of the elementary video stream in the transport stream.
audio_pid	integer	0 – 8190 (Default: **482**)	Packet Identifier (PID) of the elementary audio stream in the transport stream.
scte35_pid	integer	0 – 8190	Packet Identifier (PID) of the SCTE-35 stream in the transport stream.
timed_metadata_pid	integer	0 – 8190	Packet Identifier (PID) of the timed metadata stream in the transport stream.
private_metadata_pid	integer	0 – 8190	Packet Identifier (PID) of the private metadata stream in the transport stream.
program_num	integer	0 – 65535	The value of the program number field in the Program Map Table.
pcr_every_pes	boolean	**true** or false	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.
pcr_pid	integer	0 – 8191	Packet Identifier (PID) of the Program Clock Reference (PCR) in the transport stream.
transport_stream_id	integer	0 – 65535	The value of the transport stream ID field in the Program Map Table.
psi_repeat_rate	float	0 – 1000 (Default: **0.0**)	The number of Program Specific Information tables (PSIs) inserted in the transport stream each second. Note a value of 0 will insert once at the beginning of each segment.
audio_packets_per_pes	integer	>= 0 (Default: **16**)	The number of audio packets to insert for each PES packet.

TS SETTINGS

Requires license

NAME	TYPE	RANGE	DESCRIPTION
bitrate	integer	Default: **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.
pmt_pid	integer	0 – 8190 (Default: **480**)	Packet Identifier (PID) for the Program Map Table (PMT) in the transport stream.
video_pid	integer	0 – 8190 (Default: **481**)	Packet Identifier (PID) of the elementary video stream in the transport stream.
audio_pid	integer	0 – 8190 (Default: **482**)	Packet Identifier (PID) of the elementary audio stream in the transport stream.
scte35_pid	integer	0 – 8190	Packet Identifier (PID) of the SCTE-35 stream in the transport stream.
program_num	integer	0 – 65535	The value of the program number field in the Program Map Table.

NAME	TYPE	RANGE	DESCRIPTION
pcr_every_pes	boolean	**true** or false	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.
pcr_pid	integer	0 – 8191	Packet Identifier (PID) of the Program Clock Reference (PCR) in the transport stream.
transport_stream_id	integer	0 – 65535	The value of the transport stream ID field in the Program Map Table.
pat_repeat_rate	float	0.001 – 1000 (Default: **8.0**)	The number of Program Association Tables (PATs) inserted in the transport stream each second.
pmt_repeat_rate	float	0.001 – 1000 (Default: **8.0**)	The number of Program Map Tables (PMTs) inserted in the transport stream each second.
vbr	boolean	true or **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	true or **false**	When true, uses DVB buffer model for Dolby Digital audio. When false, the ATSC model is used.
null_packet_bitrate	float	>= **0**	Value in bits per second of extra null packets to insert into the transport stream. This can be used if a downstream encryption system requires periodic null packets.

PRESET

NAME	TYPE	RANGE	DESCRIPTION
name	string		Name for Preset.
description	string		Description for Preset.
permalink	string	Alphanumeric characters and underscores, cannot be an integer.	A short unique identifier used to refer to this Preset. For example, if the permalink is “my_preset”, it can be accessed at http://server/presets/my_preset. If left blank, a permalink will be generated based on the Preset name.
log_edit_points	boolean	true or **false**	Generates an XML file in the job log directory with initial timecode, timecode of input switches, and final timecode. This can be used to for later editing of this output.
preset_category	integer or string		Name or ID of Preset Category.
container	enum	mp4, f4v, wmv, raw, m2ts, m3u8, ismv, ts, mov, uvu	Container for this output. See Containers for supported output containers. Can be auto-detected from extension field. Certain containers require a *container*_settings object. If not specified, the default object will be created.
*container*_settings	Container Settings	mp4_settings, f4v_settings, mov_settings, uvu_settings, m2ts_settings, ts_settings	Container specific settings. Note: replace *container* with the container you are using in the XML tag (e.g. <mov_settings>).
video_description	Video Description		Video settings for this Preset.

NAME	TYPE	RANGE	DESCRIPTION
audio_description	Audio Description		Audio settings for this Preset. There can be multiple audio settings in a single Preset.
caption_description	Caption Description		Caption settings for this Preset. There can be multiple caption settings in a single Preset.
scte35_passthrough	boolean	true or **false**	If true, passes any SCTE-35 signals from the input source to this output. Only available for certain containers.
klv_passthrough	boolean	true or **false**	If true, passes any KLV data from the input source to this output. Only available for certain containers.
ebif_passthrough	boolean	true or **false**	If true, passes any EBIF data from the input source to this output. Only available for certain containers.
nielsen_id3_passthrough	boolean	true or **false**	If true, Nielsen inaudible tones for media tracking will be detected in the input audio and an equivalent ID3 tag will be inserted in the output. Only available for certain containers.

PRESET CATEGORY

NAME	TYPE	RANGE	DESCRIPTION
name	string		Name for Preset Category.

REMIX SETTINGS PRESET

NAME	TYPE	RANGE	DESCRIPTION
name	string		Name for remix settings preset.
channels_in	integer	1 – **16**	Number of input channels to be used.
channels_out	integer	1, **2**, 6	Number of output channels to be produced.
channel_mapping	XML		Remixing values to use. See example.

JOB PROFILE

NAME	TYPE	RANGE	DESCRIPTION
name	string		Name for Profile.
description	string		Description for Profile.
permalink	string	Alphanumeric characters and underscores, cannot be an integer.	A short unique identifier used to refer to this Profile. For example, if the permalink is “my_profile”, it can be accessed at http://server/profiles/my_profile. If left blank, a permalink will be generated based on the Profile name.
preroll_input	Input		Preroll media added before main input content.
postroll_input	Input		Postroll media added after main input is complete.
timecode_config	Timecode Config		Contains settings used to acquire and adjust timecode information from inputs.
priority	integer	1 – 100 (Default: **50**)	Priority indicates the order that pending Jobs will be processed. 100 is highest priority.
notification	Notification		Settings for notification on status changes.
pre_process	Pre-Process		Settings for preprocessing steps.

NAME	TYPE	RANGE	DESCRIPTION
post_process	Post-Process		Settings for postprocessing steps.
image_inserter	Image Inserter		Settings for the image inserter.
avail_blanking	Avail Blanking		Settings for ad avail blanking.
stream_assembly	Stream Assembly		A Stream assembly for this Profile. The Profile can have several stream assemblies which define output codec settings.
output_group	Output Group		An output group for this Profile. Output groups contain information about where streams should be distributed.

JOB WATCH FOLDER

NAME	TYPE	RANGE	DESCRIPTION
incoming	Location		Folder to watch for new content.
profile	Profile	Valid Profile ID, name, or permalink	Profile to use as a template for Job creation. A valid ID, name or permalink must be provided. To ensure accuracy, always use permalinks that are distinct from Profile names.
interval	integer	1 – 300 (Default: **2**)	Length of the polling interval in seconds. Increase this polling interval if Jobs are submitted before files copy.
active	boolean	**true** or false	Whether this Watch Folder is currently active.

FORMAT IDENTIFIER PARAMETERS

Certain fields allow for format identifiers to be specified that will modify the output value.

Note that when format identifiers are used in an output path, the validations preventing duplicate output paths will be disabled. If the expanded format identifiers create duplicate output paths the Job will error once it is started.

IDENTIFIER	FORMAT	DESCRIPTION
dt	YYYYMMDDTHHMMSS	UTC datetime of the start time of the Job. NOTE: HLS outputs will use the current datetime for each segment.
d	YYYYMMDD	UTC date of the start time of the Job. NOTE: HLS outputs will use the current date for each segment.
t	HHMMSS	UTC time of the start time of the Job. NOTE: HLS outputs will use the current time for each segment.
rv	Kb	Video bitrate
ra	Kb	Total of all audio bitrates
rc	Kb	Container bitrate, or the sum of video and all audio bitrates if container bitrate is not specified.
w	Pixels	Horizontal resolution
h	Pixels	Vertical resolution
f	Integer FPS	Framerate without decimal places
fn	Filename	Name of input file (excluding the extension)
$$	$	Escaped $

Fields that accept format identifier fields include:

- Destination URI (Timestamp and Filename identifiers only)
- MS Smooth Publish Point URI (Timestamp and Filename identifiers only)
- Name Modifier

WIDTH SPECIFIER PARAMETER

Format identifiers may be modified with a width parameter:

```
%0[width]
```

In the case where the value is less than the specified width, the value will be prefixed with zeros to ensure the value is equal to the width specified. If the value is greater than the specified width then the full value will be displayed (no truncation). The following example shows what happens when using the width specifier on a vertical resolution attribute of 1280:

```
$h%05$ => 01280
$h%03$ => 1280
```

SNMP INTERFACE

The Elemental Server 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 Job errors.

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

SNMP BASICS

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

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

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

returns

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

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

- http://server_ip/mib/ELEMENTAL_MIB.txt - Base MIB for all Elemental products
- http://server_ip/mib/ELEMENTAL_SERVER_MIB.txt - Objects specific to Elemental Server

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 Server SNMP interface can be queried via snmpwalk as follows:

```
snmpwalk -c elemental_snmp -v2c -m ELEMENTAL-MIB:ELEMENTAL-SERVER-MIB \
    localhost elemental
```

NOTE: On a system with a large number of saved Jobs, 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

Job status is viewed using the jobTable from the ELEMENTAL-SERVER-MIB. The jobTable provides the following variables:

VARIABLE	TYPE	GET VALUES
ELEMENTAL-SERVER-MIB::jobId	Integer	Job ID (Used as the index to the jobTable)
ELEMENTAL-SERVER-MIB::jobPending	Integer	1 if the Job is currently pending, 0 otherwise
ELEMENTAL-SERVER-MIB::jobRunning	Integer	1 if the Job is currently running, 0 otherwise
ELEMENTAL-SERVER-MIB::jobError	Integer	1 if the Job is in the error state, 0 otherwise
ELEMENTAL-SERVER-MIB::jobComplete	Integer	1 if the Job is in complete, 0 otherwise

SNMP TRAPS

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

SNMP Traps are generated for the following events:

NOTIFICATION	EVENT	CONTENTS
ELEMENTAL-MIB::alert	Any alert generated by the system	ELEMENTAL-MIB::alertSet: 1 if the alert is being set, 0 if the alert is being cleared ELEMENTAL-MIB::alertMessage: Message describing the alert that was set or cleared
ELEMENTAL-SERVER-MIB::jobRunningNotification	Job is started or stopped	ELEMENTAL-SERVER-MIB::jobId: The ID of the Job that has started or stopped ELEMENTAL-SERVER-MIB::jobRunning: 1 if the Job has started, 0 if it has completed
ELEMENTAL-SERVER-MIB::jobErrorNotification	Job had an error	ELEMENTAL-SERVER-MIB::jobId: The ID of the Job that had an error ELEMENTAL-SERVER-MIB::jobError: 1 indicating the error state. Refer to the web interface or REST API for details on the error.

AUTHENTICATION

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

MANAGING ROLES

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

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

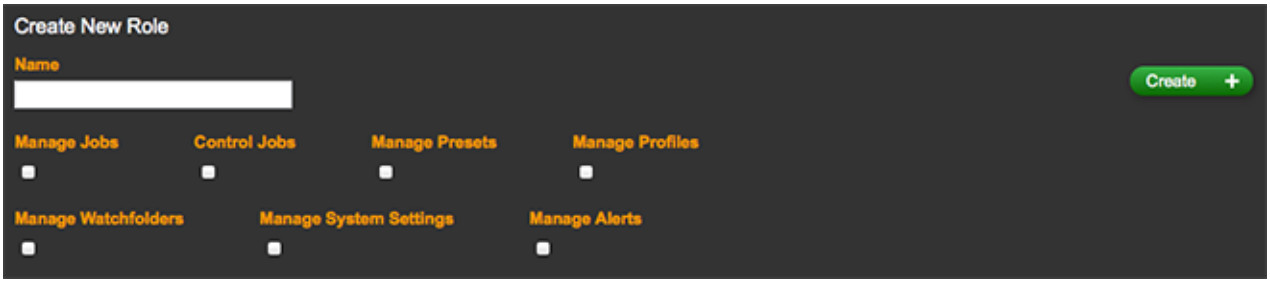
- **Admin**: The Admin role has access to the entire Elemental Server system
- **Manager**: The Manager role can create and edit Jobs, Presets, Profiles and Watch Folders, and can control Jobs
- **Operator**: The Operator role can only control Jobs (Cancel, Archive, etc.)

- **Viewer**: The Viewer role has read-only access to the Elemental Server system

CREATING NEW ROLES

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

- **Manage Jobs**: Allows user to create and edit Jobs
- **Control Jobs**: Allows user to control the state of Jobs (Cancel, Archive, etc)
- **Manage Presets**: Allows user to create and edit Presets, Preset Categories, and Audio Remixing Presets
- **Manage Profiles**: Allows user to create and edit Profiles
- **Manage Watch Folders**: Allows user to create and edit Watch Folders
- **Manage System Settings**: Allows user to update the Elemental Server system settings
- **Manage Alerts**: Allows user to update alert thresholds and to update alert notification settings



MANAGING USERS

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

CREATING NEW USERS

To create a user, the admin user must fill out the Login, Password and Password Confirmation fields, as well as select the user's Role. The 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.

SUPPORTED INPUT CONTAINERS AND CODECS

Elemental Server supports a wide range of input data containers, as well as video and audio codecs. For each container group below, there are one or more container variants each of which support the range of video codecs listed below it. The audio codecs and file extensions listed in a container group are supported by all container variants in the group.

NO CONTAINER

CONTAINERS & VIDEO CODECS

RAW

MPEG-4 part 10 video (H.264), MPEG-2

FILE EXTENSIONS

.raw, .264, .h264, .m2v

AVI

CONTAINERS & VIDEO CODECS

AVI

MPEG-4 part 2 (DivX/Xvid), DV/DVCPRO, Raw (uncompressed)

FILE EXTENSIONS

.avi, .divx, .xvid

AUDIO CODECS

Dolby Digital, Dolby Digital Plus, MPEG Audio, PCM

FLASH

CONTAINERS & VIDEO CODECS

F4V

Flash 9, MPEG-4 part 10 video (H.264)

FILE EXTENSIONS

.f4v, .flv

AUDIO CODECS

AAC, HE-AACv1, HE-AACv2

MATROSKA

CONTAINERS & VIDEO CODECS

MKV

MPEG-4 part 10 video (H.264), MPEG-4 part 2 video, MPEG-2, VC-1

FILE EXTENSIONS

.mkv

AUDIO CODECS

AAC, HE-AACv1, HE-AACv2, Dolby Digital, Dolby Digital Plus

MPEG ELEMENTARY STREAMS

CONTAINERS & VIDEO CODECS

MPEG ES

MPEG-1, MPEG-2

FILE EXTENSIONS

.m2v, .m1v

AUDIO CODECS

Dolby Digital, Dolby Digital Plus, AAC, HE-AACv1, HE-AACv2, PCM, MPEG-1 layer II, AIFF, AES3

MPEG TRANSPORT STREAMS

CONTAINERS & VIDEO CODECS

MPEG TS

MPEG-4 part 10 video (H.264), MPEG-2, ATSC (A/53)

FILE EXTENSIONS

.m2ts, .m2t, .mts, .ts, .trp, .mpeg

AUDIO CODECS

Dolby Digital, Dolby Digital Plus, AAC, HE-AACv1, HE-AACv2, PCM, MPEG-1 layer II, AIFF, AES3

MPEG-1 SYSTEM STREAMS

CONTAINERS & VIDEO CODECS

MPEG SS

MPEG-1

FILE EXTENSIONS

.mpg, .mpeg

AUDIO CODECS

Dolby Digital, Dolby Digital Plus, AAC, HE-AACv1, HE-AACv2, PCM, MPEG-1 layer II, AIFF, AES3

MPEG VIDEO FILES

CONTAINERS & VIDEO CODECS

MPEG-1 CONTAINER

MPEG-1

MPEG-2 CONTAINER

MPEG-2

FILE EXTENSIONS

.mpg, .mpeg

AUDIO CODECS

Dolby Digital, Dolby Digital Plus, AAC, HE-AACv1, HE-AACv2, PCM, MPEG-1 layer II, AIFF, AES3

MPEG-4

CONTAINERS & VIDEO CODECS

MPEG-4 CONTAINER

MPEG-4 part 10 video (H.264), MPEG-4 part 2, MPEG-2, JPEG2000, Raw (uncompressed), MJPEG, VC-1, H.261, H.262, H.263, DivX/Xvid

FILE EXTENSIONS

.mp4, .m4v, .f4v

AUDIO CODECS

Dolby Digital, Dolby Digital Plus, AAC, HE-AACv1, HE-AACv2, PCM

MATERIAL EXCHANGE FORMAT

CONTAINERS & VIDEO CODECS

MXF (OP1A)

DV25, DV50, DVCProHD, MPEG-2 I-frame, MPEG-2 Long GOP

MXF (OPATOM)

DV25, DV50, DVCProHD

MXF

Sony XDCam, Sony XDCam MPEG-4 Proxy, Panasonic P2, JPEG2000, DNxHD, VC-1, MPEG-4 part 10 video (H.264), Raw (uncompressed)

FILE EXTENSIONS

.mxf

AUDIO CODECS

PCM, AES3, MPEG-1 layer II, AIFF

QUICKTIME

CONTAINERS & VIDEO CODECS

QUICKTIME

DV/DVCPRO, MPEG-4 part 10 video (H.264), MPEG-4 part 2, MPEG-2, ProRes, JPEG2000, Raw (uncompressed), MJPEG, VC-1, H.261, H.262, H.263, DivX/Xvid

FILE EXTENSIONS

.mov

AUDIO CODECS

AAC, HE-AACv1, HE-AACv2, PCM

VIDEO OBJECT FILES

CONTAINERS & VIDEO CODECS

VOB

MPEG-2

FILE EXTENSIONS

.vob

AUDIO CODECS

Dolby Digital, Dolby Digital Plus, PCM

WMV/ASF

CONTAINERS & VIDEO CODECS

WMV/ASF

VC-1

FILE EXTENSIONS

.wmv, .asf, .ts

AUDIO CODECS

WMA, WMA2, WMA3

SUPPORTED OUTPUT CONTAINERS AND CODECS

Elemental Server can produce the following output formats. The Raw Extension column denotes what the raw essence file will be named when the Raw container is selected. For example, an output with a video codec of H.264 and an audio codec of AAC will produce one <name>.264 file and one <name>.aac file.

When selecting a codec using the REST interface, use the codec name listed in the left column. For example, to select the VC-1 codec:

```
<job>
  ...
  <media>
    <video_description>
      <codec>VC1</codec>
      ...
    </video_description>
    ...
  </media>
</job>
```

VIDEO CODECS

CODEC	DESCRIPTION	RAW EXTENSION
H.264	MPEG-4 AVC (H.264)	264
MPEG2	MPEG-2	m2v
VC1	Microsoft VC-1	vc1
Frame Capture	Frame Capture to JPEG	jpg
Uncompressed	Uncompressed	yuv
ProRes	Apple ProRes 422	prores

AUDIO CODECS

CODEC	DESCRIPTION	RAW EXTENSION
AAC	Advanced Audio Coding	aac
MP2	MPEG-1 Layer II	mp2
WMA2	Windows Media Audio WMA2	wma
WAV	Uncompressed WAV	wav
AIFF	Uncompressed AIFF	aiff
AC3	Dolby Digital	ac3
Pass Through	Dolby Digital Pass Through	
EAC3	Dolby Digital Plus	ec3
DTSE	DTS Express	dtse

The following codecs and settings are supported with the Pass Through audio codec.

CODEC	CHANNELS	CODING MODE	SAMPLE RATES	BITRATES(KBPS)
ac3	1	1_0	32k, 44.1k, 48k	56, 64, 80, 96, 112, 128, 160, 192
ac3	2	2_0	32k, 44.1k, 48k	96, 112, 128, 160, 192, 224, 256, 320, 384
ac3	6	3_2 + LFE	32k, 44.1k, 48k	224, 256, 320, 384, 448, 512, 576, 640

CONTAINERS

When selecting a container using the REST interface, use the container's default extension. For example, to select the MPEG-4 container:

```
<job>
  ...
  <media>
    <container>mp4</container>
    ...
  </media>
</job>
```

The following combinations of Containers, Video Codecs, and Audio Codecs are supported:

CONTAINER	DEFAULT EXTENSION	VIDEO CODECS	AUDIO CODECS
MPEG-4 Container	mp4	H.264	Pass Through, AAC, AC3, EAC3, DTSE
MPEG-4 Flash Container	f4v	H.264, MPEG2	AAC
Microsoft WMV/ASF	wmv	VC1	WMA2
Raw (No container)	raw	H.264, VC1, Frame Capture, MPEG2, Uncompressed	AAC, WAV, AC3, AIFF, MP2, DTSE, EAC3
MPEG-2 Transport Stream	m2ts	H.264, MPEG2	AAC, Pass Through, MP2, AC3, EAC3
MPEG-2 Transport Stream for Apple Adaptive	m3u8	H.264	AAC, AC3, EAC3, Pass Through
ISMV Container for Microsoft Smooth Streaming	ismv	H.264, VC1	AAC, WMA2, AC3, EAC3, Pass Through
MPEG-2 TS (CableLabs)	ts	H.264, MPEG2	Pass Through, AAC, AC3, EAC3, MP2
Quicktime	mov	H.264, MPEG2, ProRes	Pass Through, AAC, AC3, EAC3, WAV, AIFF
Ultraviolet	uvu	H.264	AAC, AC3, EAC3, DTSE, Pass Through
HDS	f4m	H.264	AAC