AWS ELEMENTAL CONDUCTOR FILE

API REFERENCE

2.13.1 RELEASE

DECEMBER 5, 2019

TABLE OF CONTENTS

- Overview
- Web Interface
- REST Interface
- Elemental/Server/Job Parameters
- SNMP Interface
- Authentication
- Reference
  - Supported Codecs
  - Supported Captions

PURPOSE

- Purpose
- Overview
- Node Control
- Jobs
- Job Search
- Presets
- Job Profiles
- Job Watch Folders
- Notifications
- Statistics
- Advanced Pre and Post Processing
    - Pre Processing
    - Post Processing
    - Custom Scripts
- Errors and Warnings
- Troubleshooting

This document is intended for system integrators and users of AWS Elemental Conductor File. 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

AWS Elemental Conductor File 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.

AWS Elemental Conductor File 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 AWS Elemental Conductor File. 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 AWS Elemental Conductor File 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 AWS Elemental Conductor File system or for diagnostics.

NODE CONTROL

AWS Elemental Conductor File will search your Local Area Network for any AWS Elemental Server systems. When it detects an AWS Elemental Server system, it will appear on the Nodes list on the left-hand side of page in the Discovered state. The operator must then decide if they want to add this node to the cluster being managed by AWS Elemental Conductor File, or leave this node as a standalone encoding node.

The autodetection of AWS Elemental Server nodes relies on Zeroconf to be functional on the network. Local-area networks that do not support Multicast traffic or otherwise block Zeroconf traffic may require the operator to manually add AWS Elemental Server nodes to the cluster. This may be accomplished using the Web Interface by selecting 'Discover Node' under the Nodes tab in the navigation and entering the hostname or IP address of the node in the modal. Alternatively, POST to http://<server>/nodes/discover using the <host>name_or_ip</host> parameter. Once added to the cluster, the Conductor system will be capable of configuring an AWS Elemental Server node. Nodes that already appear in the Nodes index may still be "discovered", but will remain in whatever state currently displayed (e.g. "Standalone", or "Active").

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 Supported Codecs for a list of valid input media and supported output formats.

Transcoding Jobs are prioritized, submitted, distributed, queried, and deleted through the interfaces described in this document. AWS Elemental Conductor File'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 Queue 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. AWS Elemental maintains a list of common Presets that are delivered to the AWS Elemental Conductor File system via software updates. Additionally, the user can specify named Presets using any of the interfaces to the AWS Elemental Conductor File 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 AWS Elemental Conductor File 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 AWS Elemental Conductor File and associated with a specified Job Profile. The folders are monitored by AWS Elemental Conductor File for new input media files. When a new file is placed in a folder that is associated with a Job Watch Folder, AWS Elemental Conductor File 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

AWS Elemental Conductor File 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.

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 AWS Elemental Conductor File 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:

File handling for this job: This feature will control file handling on this job, possibly overriding the global progressive reader setting. The settings are:

- Use Global Progressive Reader Setting - files will be handled according to the global setting.
- Enable Progressive Reader and do not copy to local - S3, HTTP and networked files will be read in place.
- Disable Progressive Reader and copy to local - all files will be copied locally prior to processing. Use this setting when network integrity cannot be guaranteed during processing.
- Copy to local only from S3 and HTTP location - disable Progressive Reader, but still read networked files in place.

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 AWS Elemental Conductor File may be diagnosed by viewing the log files available here: http://server_ip/logs.

For additional support, contact AWS Elemental Support through your company's Private Space on the AWS Elemental User Community.

WEB INTERFACE QUICK START GUIDE

- Terms
- Icons
- Navigation
- Node Manager
- Getting Started
- Nodes
  - Configure an AWS Elemental Server Node
- Jobs
  - Creating a Job
  - File Groups
  - Apple Live Groups
  - DASH ISO Groups
  - Adobe HDS Groups
  - Microsoft Smooth Groups
- Creating a Job from XML or Profile
- Saving and Managing a Job
- Advanced Audio Track Selections
- Presets
- Profiles
- Watch Folders
- Stats
- Conductor Configuration
  - Node Configuration

WEB INTERFACE QUICK START GUIDE

AWS Elemental Conductor File provides a web interface for fast, effective operation of a cluster of AWS Elemental Server encoders. This page explains the basic steps for using the default web interface and defines the terms used in the interface.

DEFINITION OF COMMON TERMS

- **Node:** an AWS Elemental Server unit that can be managed by AWS Elemental Conductor
  - How to manage a Node
- **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 Queue page screenshot.

🔍
Show:
This icon indicates that more information is available about the given object. For example, this icon is used on the Job Queue 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 AWS Elemental Conductor File.

⧉
Duplicate:
This icon indicates the duplication of an object. It is used on the Job Queue 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 AWS Elemental Conductor File 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 Queue page. Archived Jobs can be found by clicking the Archive filter button on the Job Queue page.

Icons are also used on the Nodes page, but as the actions performed by the icons are very different from Jobs, they are listed separately here.

🔍
Show:
Links to the Node Show page

🔧
Config:
This icon links to the Node Configuration page.

✚
Add to Cluster:
This icon adds the node to the cluster. The AWS Elemental Conductor File node will run the configuration script on the node. It is suggested to edit the Node Configuration settings before adding a node to a cluster.

▶
Activate:
This icon is used to activate waiting nodes so that they rejoin the cluster.

📢
Force Failure:
This icon is used to force a node to fail.

🔨
Reconfigure:

This icon is used to re-run the configuration script on a failed node.

⚡
Upgrade Software
This icon is used to upgrade the software version on a node. If the node is not part of the cluster this action will also add it to the cluster.

↻
Restart elemental_se Service:
This icon is used to restart the elemental_se service on a node. This action will error any running Jobs on that node.

↪
Remove from Cluster:
This icon is used to remove a node from the Conductor cluster. This action will error any running Jobs on that node.

There are 7 base pages for the default web interface

- Nodes: View status of nodes in the cluster, configure nodes, view statistics on nodes
- Job Queue: View status of current Jobs, current and past Job details, or create new Job
- Profiles: View, create and edit Job Profiles
  - Presets: View, create and edit Presets, Preset Categories, and Audio Remixing Presets
- Job Watch Folders: View, create and edit Job Watch Folders
- Stats: Provides statistics for the AWS Elemental Conductor File system
- Conductor Configuration: Modify Conductor settings
  - Node Configuration: Modify individual node settings
- Support: Documentation for the web interface, the REST interface, and the SNMP API

USING THE NODE MANAGER

The Node Manager is located on every page and contains a list of all the nodes within the cluster and displays each node's status.

On most pages the Node Manager is hidden by default. To display the Node Manager, click the grey Node tab located on the leftmost edge of the page.

When a node is in the 'Waiting' state, the Node Manager provides a '+' button to easily add it back to the cluster without having to navigate away from the current page. The Node Manager also has another very useful task. It is used to facilitate navigation and filtering on the various pages in order to more quickly access information and settings that are important in each area.

Next to each node in the Node Manager, there is a '>' button. Clicking this button performs the filtering or navigation action for that node. The following list describes which action will be taken on each page. Hovering over the '>' button will also give a pop-up tooltip describing the action.

- **Nodes:** Navigates to the Node Show page
- **Jobs:** Filters the Jobs displayed by node
- **Profiles and Presets:** No Effect
- **Diagnostics:** Filters the alerts, messages by node
  - **Logs:** Displays the log files on a node
  - **Node Update Logs:** Displays the log files the Conductor node creates when updating a node
- **Conductor and Node Configuration Pages:** Navigates to the Node Configuration page for that node
- **Support:** No Effect

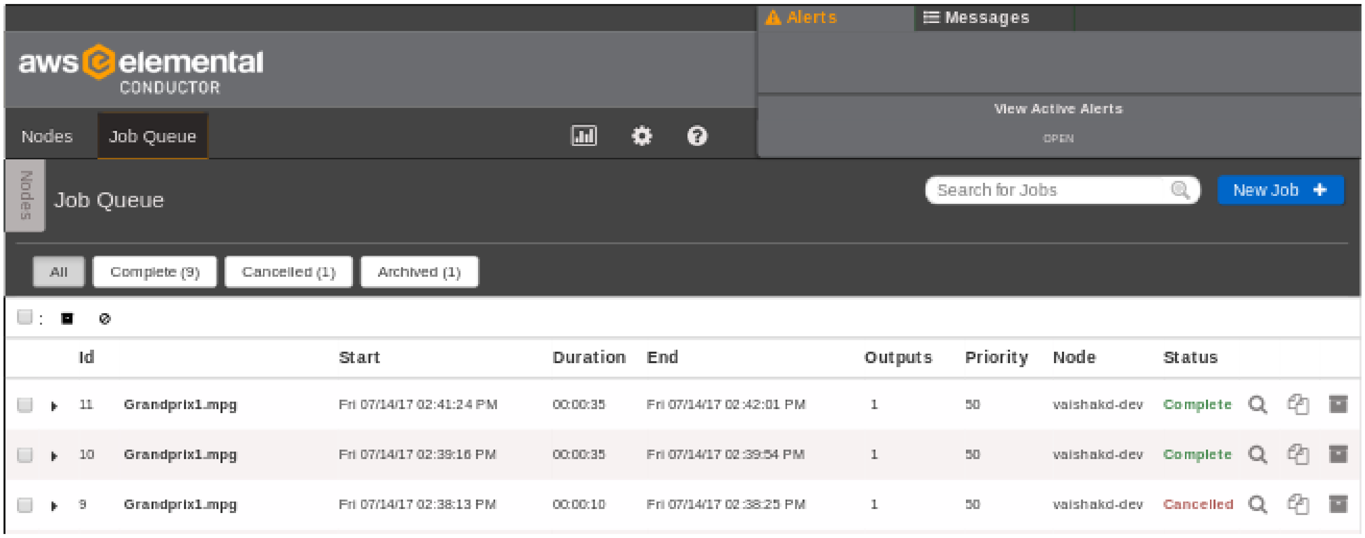
In order to remove any node filtering or navigate back to the main page for that section, there will be an entry in the Node Manager at the top with another '>' button and a description of the action that will be performed.

TYPICAL STEPS FOR GETTING STARTED WITH AWS ELEMENTAL CONDUCTOR FILE

Point a web browser at the AWS Elemental Conductor File web address:

http://<ip address of server>

You should see a screen like this:

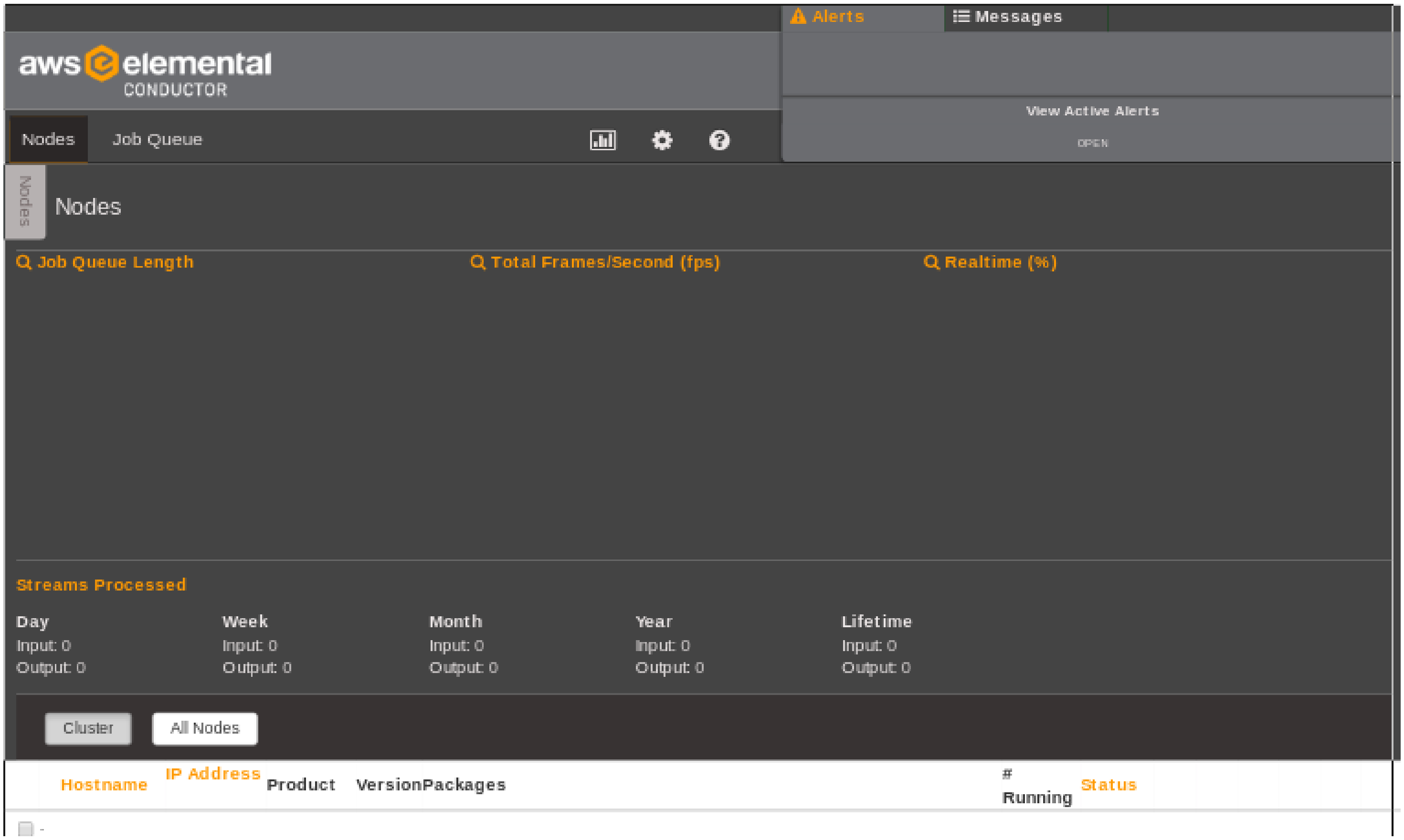


		Id		Start	Duration	End	Outputs	Priority	Node	Status
☐	▸	11	**Grandprix1.mpg**	Fri 07/14/17 02:41:24 PM	00:00:35	Fri 07/14/17 02:42:01 PM	1	50	vaishakd-dev	Complete
☐	▸	10	**Grandprix1.mpg**	Fri 07/14/17 02:39:16 PM	00:00:35	Fri 07/14/17 02:39:54 PM	1	50	vaishakd-dev	Complete
☐	▸	9	**Grandprix1.mpg**	Fri 07/14/17 02:38:13 PM	00:00:10	Fri 07/14/17 02:38:25 PM	1	50	vaishakd-dev	Cancelled

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

NODES PAGE

The *Nodes* page will contain a list of all the AWS Elemental Server nodes that the Conductor system has detected on its local area network. If there is an active AWS Elemental Server node that has not been detected, that node can be manually configured to join the Conductor cluster using the installer on the node.



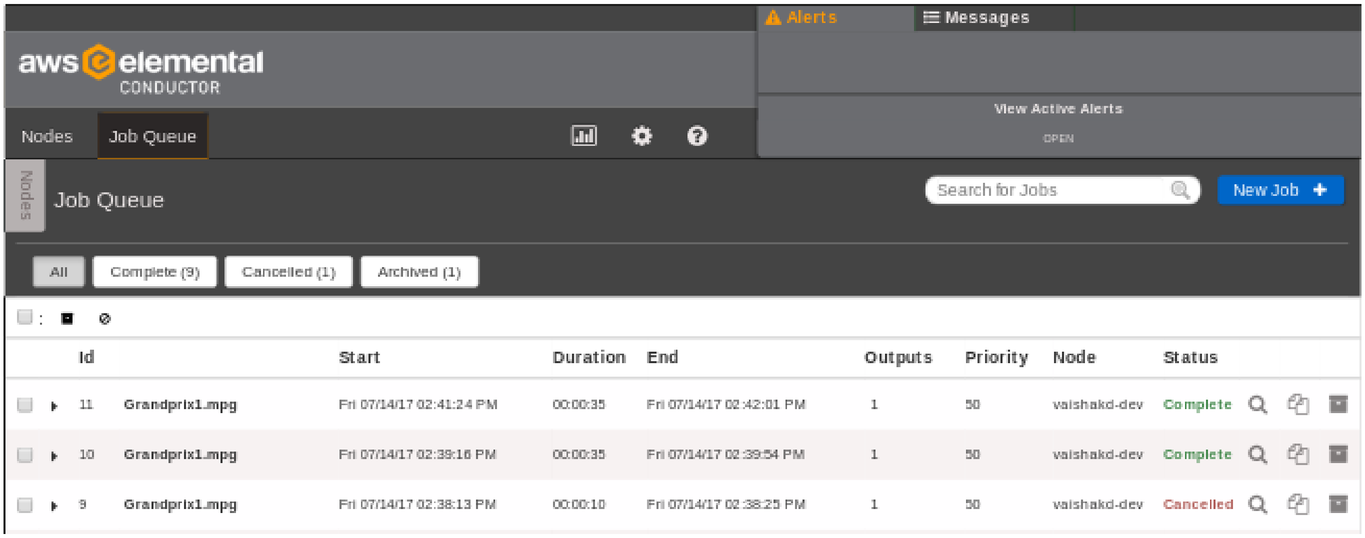
CONFIGURING AN AWS ELEMENTAL SERVER NODE

Nodes that have been detected will appear in the Nodes list in the "Discovered" state. The operator can choose "Add to Cluster" to make this node available to run Jobs controlled by the Conductor. If the user wishes for this node to act as a standalone AWS Elemental Server system, they can select "Remove from Cluster". When a node is a member of the cluster, its network, firewall, and mount point configuration can be controlled from the node's Settings page.

When a node is added to the cluster, the Conductor system will run the configuration script on that node. A log file will be created on the Conductor system with the output of that script. If there are any problems with the configuration, consult this log file or contact AWS Elemental support. Contact AWS Elemental Support through your company's Private Space on the AWS Elemental User Community.

JOB QUEUE

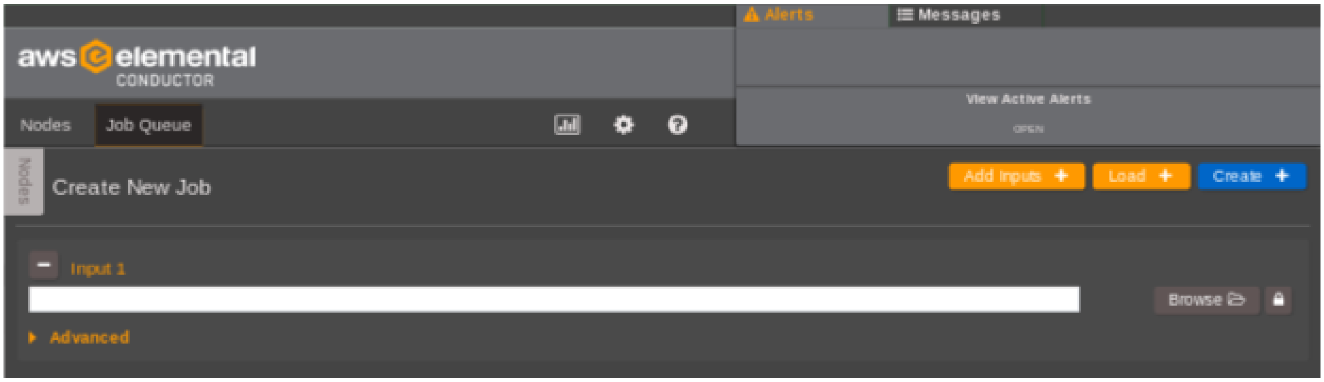
The *Job Queue* page contains a list of all the Jobs on the Conductor system and provides controls to create, cancel, and duplicate Jobs, among other things.



	Id		Start	Duration	End	Outputs	Priority	Node	Status
☐	11	Grandprix1.mpg	Fri 07/14/17 02:41:24 PM	00:00:35	Fri 07/14/17 02:42:01 PM	1	50	vaishakd-dev	Complete
☐	10	Grandprix1.mpg	Fri 07/14/17 02:39:16 PM	00:00:35	Fri 07/14/17 02:39:54 PM	1	50	vaishakd-dev	Complete
☐	9	Grandprix1.mpg	Fri 07/14/17 02:38:13 PM	00:00:10	Fri 07/14/17 02:38:25 PM	1	50	vaishakd-dev	Cancelled

CREATING A NEW JOB

After selecting the *Job Queue* 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.

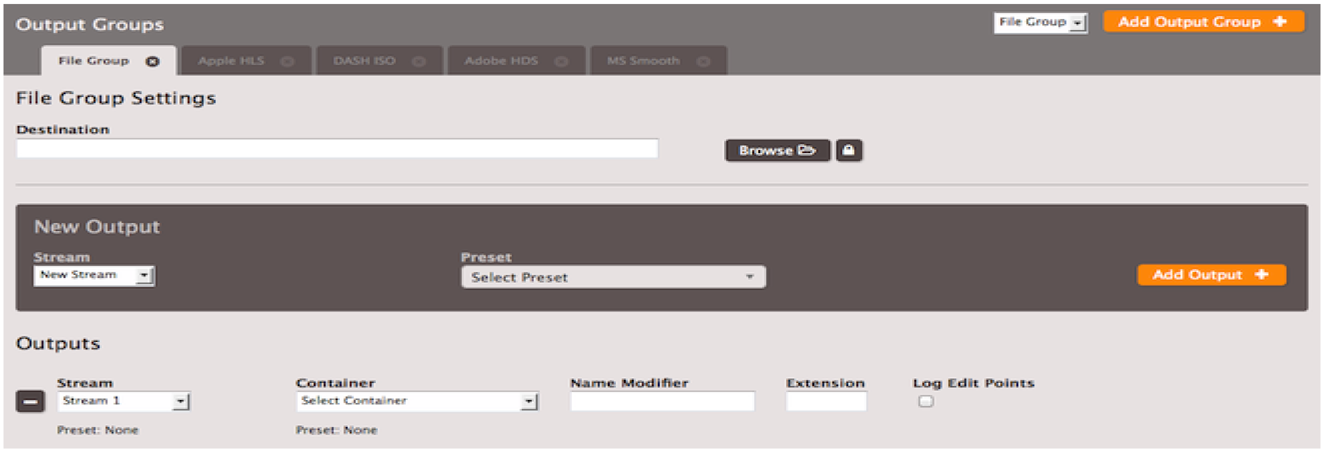
CONFIGURING OUTPUTS

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.

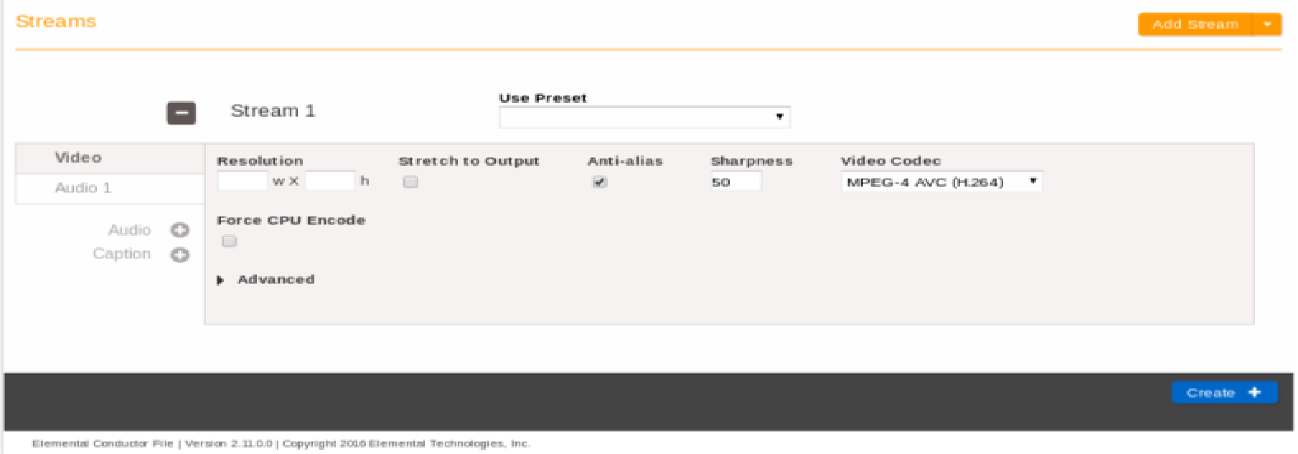


New Output: Stream: New Stream; Preset: Select Preset

Outputs: Stream: Stream 1 (Preset: None); Container: Select Container (Preset: None)

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.



Anti-alias: ☑; Sharpness: 50; Video Codec: MPEG-4 AVC (H.264)

Elemental Conductor File | Version 2.11.0.0 | Copyright 2016 Elemental Technologies, Inc.

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 LIVE GROUP

Global Apple Live parameters found under the Apple Live 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 Live 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 Live group must have video.

SETTING UP A DASH ISO GROUP

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

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

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.

CREATING A JOB FROM AN XML RESOURCE OR A PROFILE

There is an orange button labeled 'Load' at the top right of the page next to the 'Create' button. Clicking on the button will present two choices:

- Job XML
- Job Profile

Selecting Job XML will display a file browse form that allows you to select an XML file to populate the Job page. XML submitted through this interface must be valid and conform to the current system version.

Selecting Job Profile will display a drop-down list of all available Profiles. Applying a profile from the list will reload the Job page with the settings from the selected profile.

Please note that loading either one of these resources will overwrite any user data currently on the page. Both of these forms are also accessible by their own endpoints /new_from_xml and /new_from_profile.

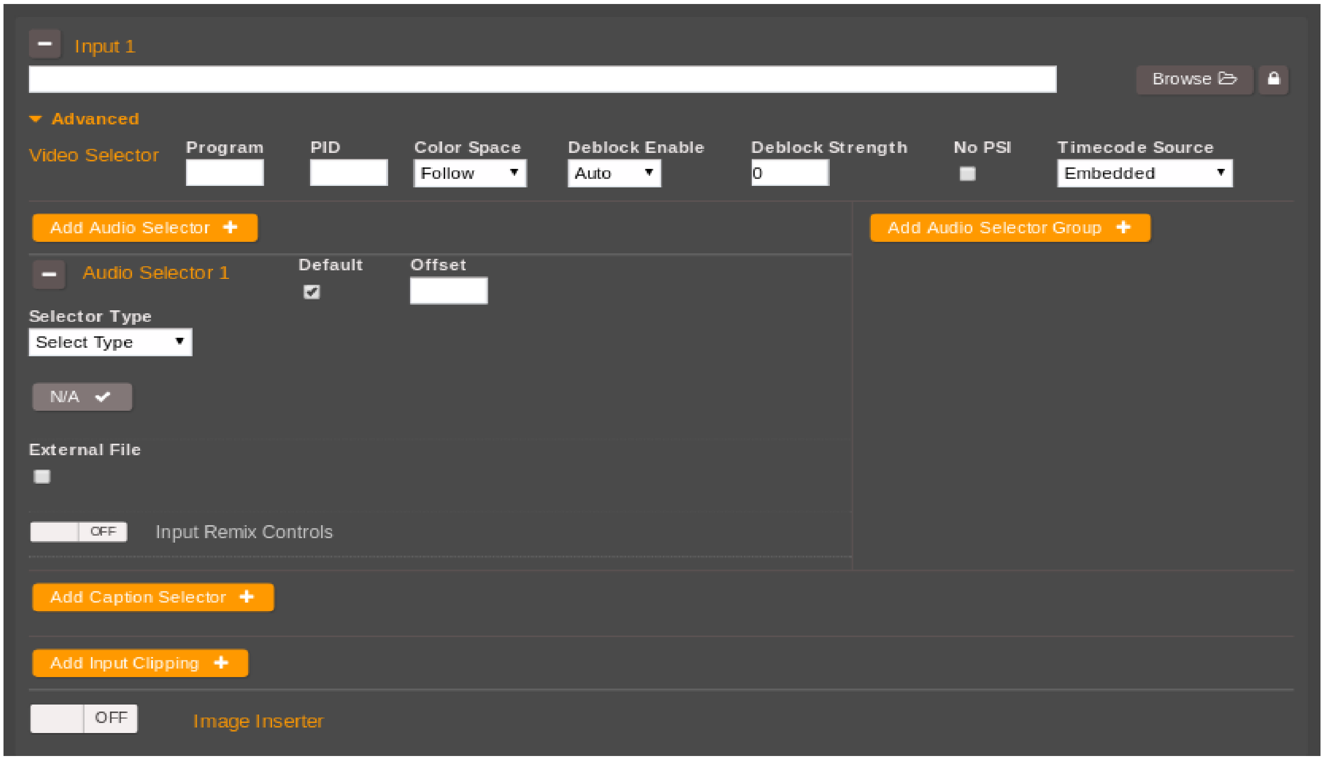
SAVING AND MANAGING A JOB

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.

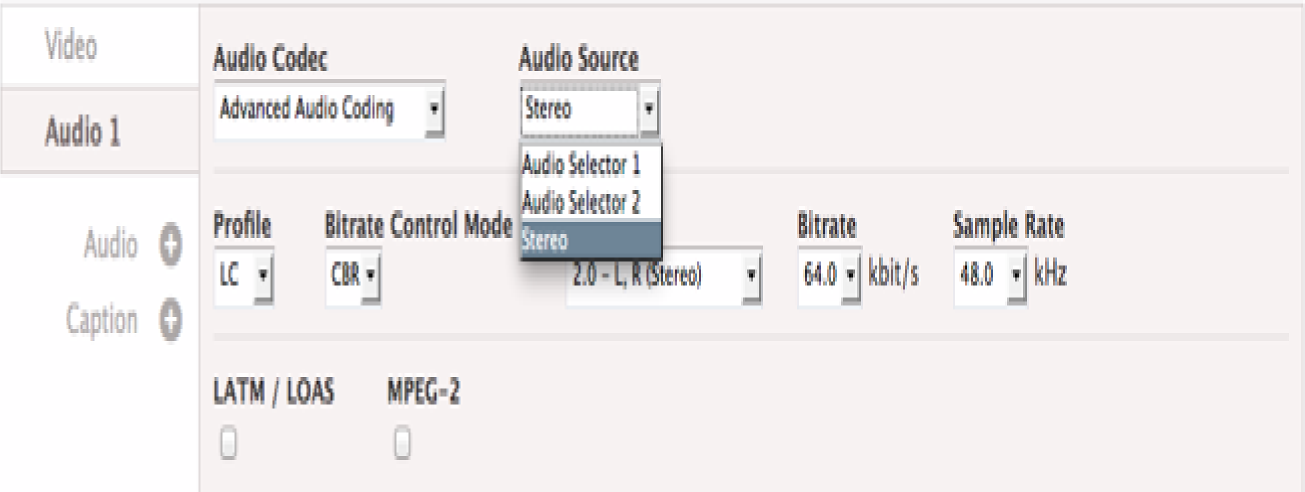
ADVANCED AUDIO TRACK SELECTIONS

AWS Elemental Conductor File 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":



Color Space: Follow
Deblock Enable: Auto
Deblock Strength: 0
Timecode Source: Embedded
Default: ☑
Selector Type: Select Type

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



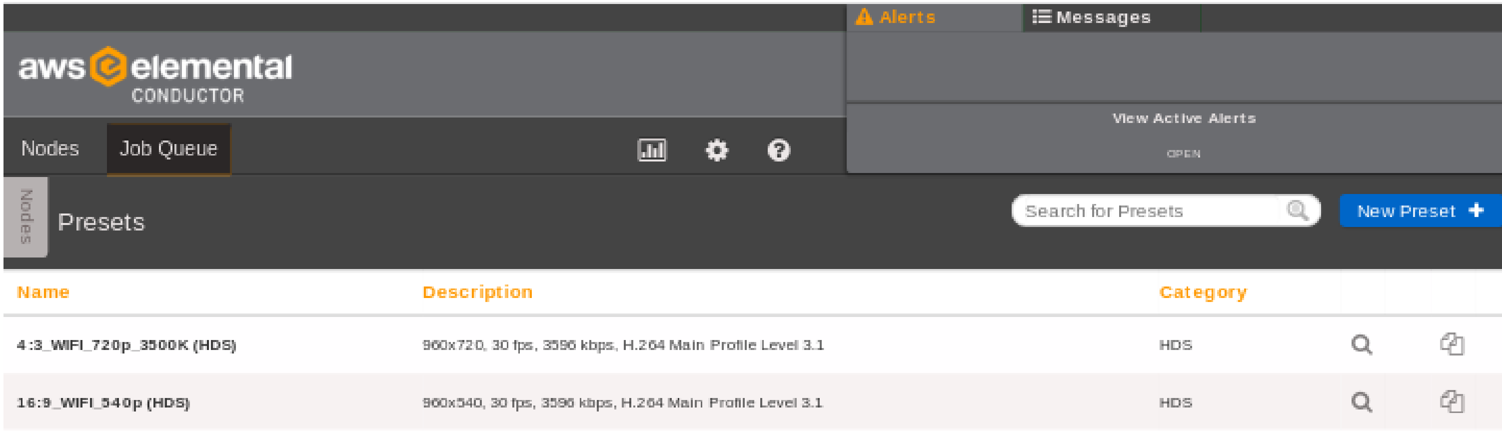
Audio Codec: Advanced Audio Coding
Audio Source: Stereo
Profile: LC
Bitrate Control Mode: CBR
2.0 - L, R (Stereo)
Bitrate: 64.0 kbit/s
Sample Rate: 48.0 kHz

For more information, see Audio Selector

USING PRESETS

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

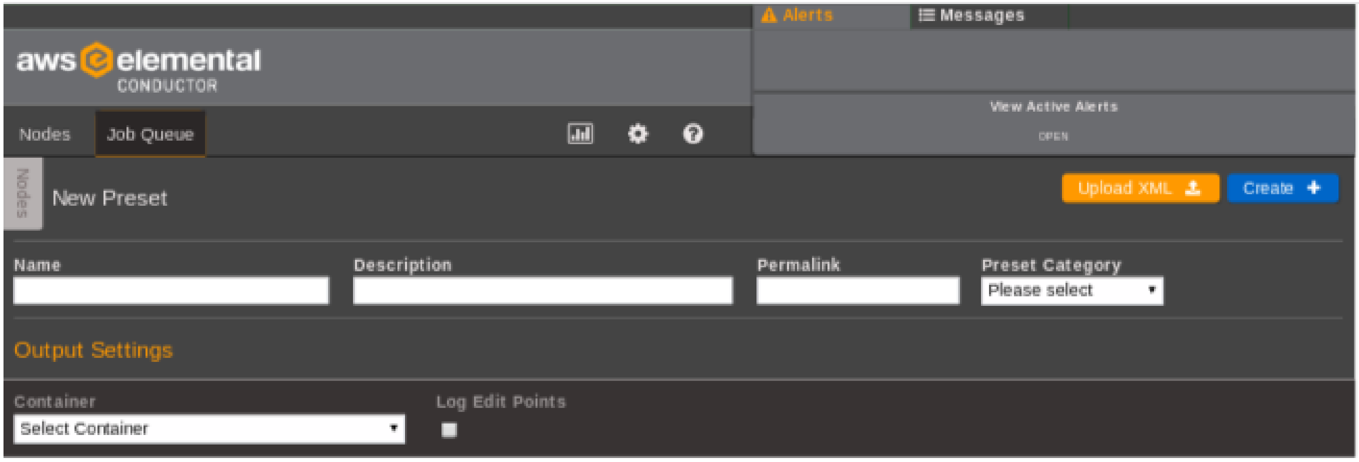
Creating a Preset is not required, but if you plan to use the same encoding information multiple times, it is recommended. Click the *Presets* tab to view existing Presets and to create new Presets.



Name	Description	Category
4:3_WIFI_720p_3500K (HDS)	960x720, 30 fps, 3596 kbps, H.264 Main Profile Level 3.1	HDS
16:9_WIFI_540p (HDS)	960x540, 30 fps, 3596 kbps, H.264 Main Profile Level 3.1	HDS

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

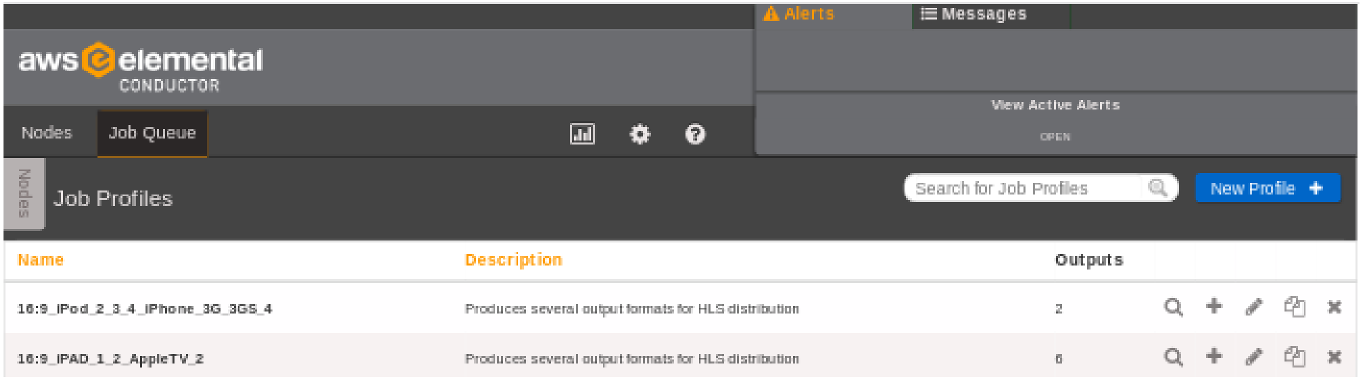
Creating a Preset using the Stream settings of an existing Job is useful if these settings will be used again. Navigate to the "Show" page for the Job and find the Stream that you wish to save as a preset. Next to the Stream label is a "Save as Preset" button. Clicking this button brings up fields for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



Name	Description	Outputs
16:9_iPod_2_3_4_iPhone_3G_3GS_4	Produces several output formats for HLS distribution	2
16:9_iPAD_1_2_AppleTV_2	Produces several output formats for HLS distribution	6

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.

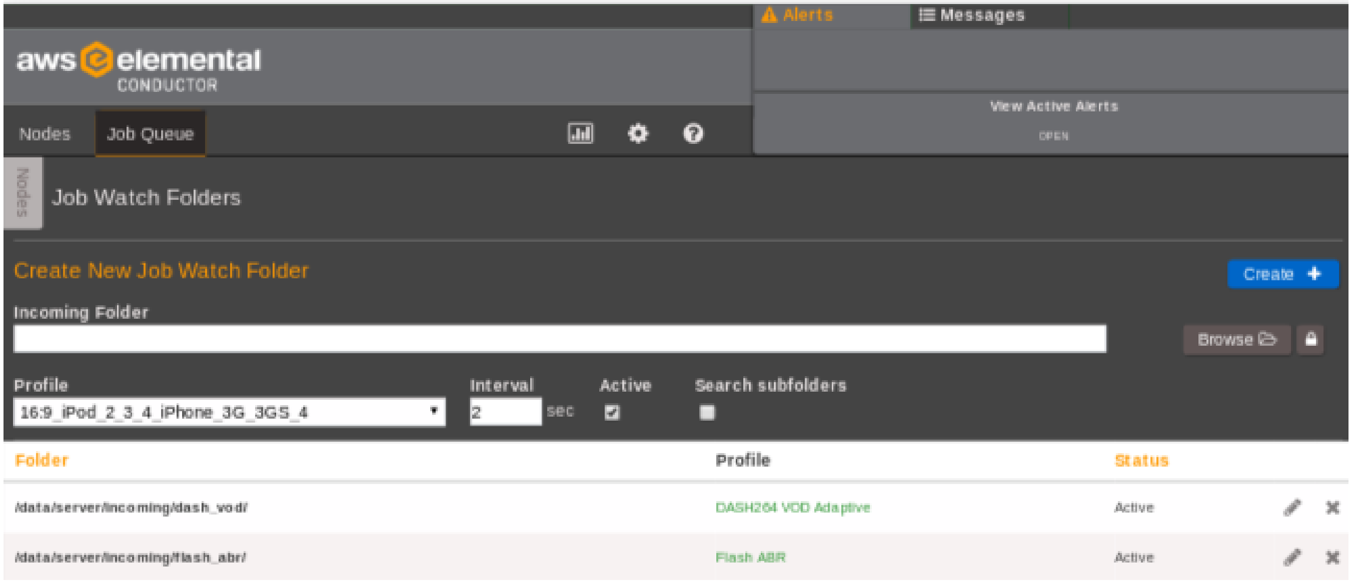
CREATING A NEW JOB PROFILE

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.



Incoming Folder

Profile: 16:9_iPod_2_3_4_iPhone_3G_3GS_4

Interval: 2 sec

Active: ☑

Search subfolders: ☐

Folder	Profile	Status
/data/server/incoming/dash_vod/	DASH264 VOD Adaptive	Active
/data/server/incoming/flash_abr/	Flash ABR	Active

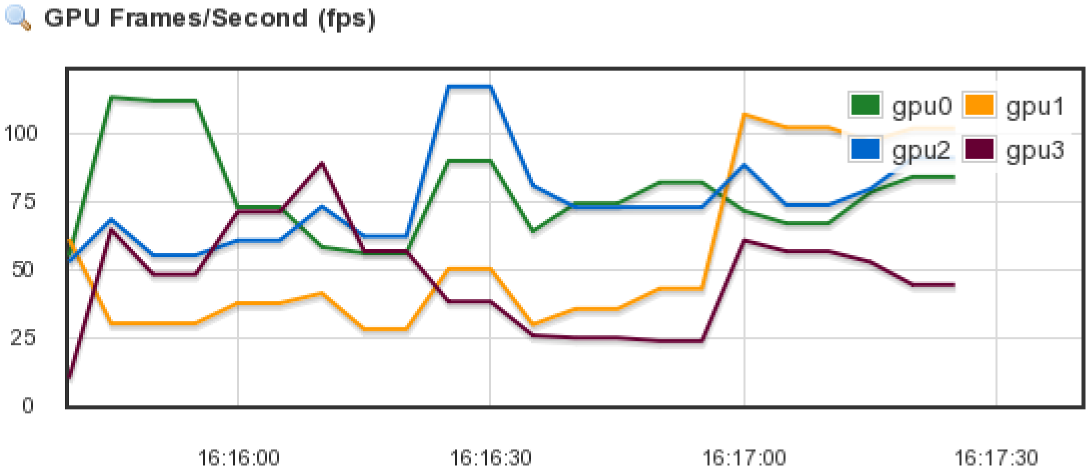
USING THE STATS PAGE

The stats pages provide various statistics about the AWS Elemental Conductor File cluster. 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 AWS Elemental Conductor File 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.

Nodes

	Node Name	Status
▼	snowwhite	Active

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.

DIAGNOSTICS

USING THE ALERTS PAGE

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

Alerts can be generated for system level events including:

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

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

USING THE MESSAGES PAGE

The Messages page is accessible from a dropdown menu on the Diagnostics tab. Messages log activity on the cluster. Messages indicate when nodes enter or exit the system, and when nodes fail or Jobs have errors, among other things. Messages can be useful to help diagnose issues within the cluster by marking the sequence of Jobs with specific timestamps.

USING THE CONDUCTOR CONFIGURATION PAGE

The Conductor Configuration page provides access to a variety of configuration options for AWS Elemental Conductor.

GENERAL SETTINGS

The General Settings page allows for selecting the timezone for all the AWS Elemental nodes, and also provides an option for disabling the browser warning that appears on unsupported browsers. AWS Elemental suggests setting the timezone before adding any nodes to the cluster . If the timezone is updated AWS Elemental suggests restarting the service on the Conductor node and all clustered nodes, and recreating any Schedules. 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. Auto-archive will remove completed Jobs from the displayed Jobs list. Auto-delete permanently removes Jobs from the system. Setting these values to "0" will disable this functionality.

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

To restore an automatic backup file called `elemental-db-backup_Conductor_File_2.13.1.0_2014-09-11_05-13-04.tar`, in the backup location `/home/elemental/database_backups`:

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

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 AWS Elemental system.

FAILOVER SETTINGS

The Failover Settings page allows for adjustment of the parameters that the Conductor system uses to determine if a failover event has occurred. The Heartbeat Frequency indicates how frequently all of the clustered nodes send a heartbeat command to the Conductor system to indicate that they are still alive. The Failover Threshold indicates how much time must elapse between heartbeats before the Conductor system initiates a failover for a node. This value should be a few seconds larger than the Heartbeat Frequency to be able to deal with any network latencies. The Automatic Rejoin checkbox causes failed nodes to automatically rejoin the cluster when they recover. If unchecked, the node will first enter the 'Waiting to Rejoin' state and will require user-input before it rejoins the cluster. This can allow on-site engineers to verify that the node is running correctly after the failure.

POOLED LICENSING

Pooled Licensing provides a set of licenses to serve any or all nodes in a cluster via AWS Elemental Conductor File. To use Pooled Licensing, upload a valid pooled license to Settings -> Licensing. The license file is placed at /opt/elemental-lics/licenses. Any newly added worker nodes without an eme.lic file will automatically apply for a license from the pool.

To use an existing local eme.lic for a worker node, no changes are needed. To use pooled licensing for a worker node not yet participating in the pool, remove eme.lic from /opt/elemental_se locally, then reconfigure the node manually through the Force Reconfigure icon on the Conductor nodes view.

Starting at 60 days prior to expiration, AWS Elemental Conductor File will display alerts pertaining to all nodes participating in the pool.

AUTHENTICATION SETTINGS

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

CLUSTER MOUNT POINT SETTINGS

The Cluster Mount Point Settings page provides status information on the active cluster mount points and provides the ability to globally add new CIFS, NFS, or DAVFS mount points to all worker nodes in an AWS Elemental Conductor File system cluster. Mount points are limited to the /data/mnt directory.

Cluster scope mount points can only be mounted, unmounted, and deleted through the cluster mount point settings page. A cluster mount point cannot be created if any cluster- or local scope mount point has the same mount folder name. Changes to local scope mount points can be made on the Node Mount Point Settings page.

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.

VERSION MANAGEMENT

The Conductor system can manage AWS Elemental software version files and use these files to upgrade any clustered nodes. The Version Management page shows which software versions are installed and allows new software versions to be uploaded to the Conductor system.

CLUSTER SEQUENCER SETTINGS

The Cluster Sequencer Settings page provides access to settings that fine-tuning the video transcoding sequencer for multiple nodes in a cluster.

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.

Separate cluster-wide settings are defined for AWS Elemental Server nodes in the cluster. If these settings are changed, all nodes subsequently added to the cluster will be synced to them by default.

Settings will be applied to nodes that are not currently running Jobs, or will set a node that is running Jobs to a "delayed shutdown" state, so that they can be applied once the Jobs is complete.

The "Save to Synced Nodes" only applies settings to nodes that are already identified as "synced", as well as nodes added to the cluster in the future. The "Save to All" action applies changes to all nodes fitting the settings description. Applying settings changes will cause nodes to become temporarily inactive while they are reconfigured. Please allow a few minutes for the settings to be applied to the system.

The "Restore Defaults" button restores the form fields with the system's original cluster sequencer settings. To keep these settings, select the "Save" button, or make additional adjustments before saving.

ROUTER SETTINGS

The Routers page allows for the configuration of SDI input routers. First, click "New Router" to set up a new router. Then fill out settings for the router's name, IP address, number of inputs and outputs, and router type. Harris Panacea routers can share an IP address, but require a unique level ID. Next, click "Apply" to customize naming for the inputs, and identify the devices connected to the router. Click "Create" to save the router.

USING THE NODE CONFIGURATION PAGE

The Node Configuration page provides access to a variety of configuration options for each AWS Elemental Server node in the cluster.

The Node Configuration page can be accessed from the Nodes page by clicking on the gear icon next to the node, or from the Conductor Configuration page by using the Node Manager.

NETWORK SETTINGS

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

CURRENT SETTINGS

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

HOSTNAME, DNS, & NTP

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

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

NETWORK DEVICES

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

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

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

Below are settings unique to device types:

eth (ethN)

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

VLAN (ethN.M)

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

bond (bondN)

Bond ID - Must be an integer.

- Mode (for Port Bonding) - Select one of the following entries.
  - *0: Round Robin* - Sets a round-robin policy for fault tolerance and load balancing. Transmissions are received and sent out sequentially on each bonded slave interface beginning with the first one available.
  - *1: Active Backup* - Sets an active-backup policy for fault tolerance. Transmissions are received and sent out via the first available bonded slave interface. Another bonded slave interface is only used if the active bonded slave interface fails.

  - *2: Balanced XOR* - Sets an XOR (exclusive-or) policy for fault tolerance and load balancing. Using this method, the interface matches up the incoming request's MAC address with the MAC address for one of the slave NICs. Once this link is established, transmissions are sent out sequentially beginning with the first available interface.
  - *3: Broadcast* - Sets a broadcast policy for fault tolerance. All transmissions are sent on all slave interfaces.
  - *4: IEEE 803.ad Dynamic Link Aggregation* - Sets an IEEE 802.3ad dynamic link aggregation policy. Creates aggregation groups that share the same speed and duplex settings. Transmits and receives on all slaves in the active aggregator. Requires a switch that is 802.3ad compliant.
  - *5: Adaptive Transmit Load Balancing* - Sets a Transmit Load Balancing (TLB) policy for fault tolerance and load balancing. The outgoing traffic is distributed according to the current load on each slave interface. Incoming traffic is received by the current slave. If the receiving slave fails, another slave takes over the MAC address of the failed slave.
  - *6: Adaptive Load Balancing* - Sets an Active Load Balancing (ALB) policy for fault tolerance and load balancing. Includes transmit and receive load balancing for IPV4 traffic. Receive load balancing is achieved through ARP negotiation.
- *Link Mode* - Select whether to use MII or ARP link monitoring for all slaves in the bond. MII is generally preferred over ARP.
- *MII Monitoring Frequency* - Specifies the MII link monitoring frequency in milliseconds. This determines how often the link state of each slave is inspected for link failures. 100ms is a good starting point.
- *Use Carrier* - Used in conjunction with the MII Link Mode. If "Use Carrier" is selected then MII will use MII or ETHTOOL ioctls (less efficient, and uses deprecated kernel calling sequences), instead of netif_carrier_ok. Relies on the device driver to maintain link state.
- *Down Delay* - Specifies the time, in milliseconds, to wait before disabling a slave after a link failure has been detected. Only applies to the MII Link Mode, and should be a multiple of the MII Monitoring Frequency (will be rounded to nearest multiple). Defaults to 0.
- *Up Delay* - Specifies the time, in milliseconds, to wait before enabling a slave after a link recovery has been detected. Only applies to the MII Link Mode, and should be a multiple of the MII Monitoring Frequency (will be rounded to the nearest multiple). Defaults to 0.
- *ARP Interval* - Specifies the ARP link monitoring frequency in milliseconds. Periodically checks slave devices for traffic, generates regular interval traffic via ARP probes for ARP IP Target.
- *ARP IP Target* - Specifies the IP address to use for ARP probes in ARP Link Mode.

RESTORE DEFAULTS

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

NODE MOUNT POINT SETTINGS

The Node Mount Point Settings page provides status information on active cluster and local scope mount points on a specific node and provides the ability to add new CIFS, NFS, or DAVFS mount points to a node in the AWS Elemental Conductor File system. Mount points are limited to the /data/mnt directory.

Cluster scope mount points can only be mounted, unmounted, and deleted through the Cluster Mount Point Settings page. Correspondingly, local scope mount points can only be modified on the Node Mount Point Settings page.

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 AWS Elemental Conductor File. 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.

INPUT DEVICES

The Input Devices page displays devices currently available to the system, and provides the ability to customize the device names as they are viewed within the system.

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.

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 advanced settings. Adjustments may then be made to the default settings. In order to commit these changes, the "Save" button must be pressed.

UPGRADE SOFTWARE

The Conductor system can be used to upgrade the software on any node. Any software versions shown on the Version Management page that match the type of software installed on the node can be selected. When using the Conductor system to upgrade software, an **upgrade_software.log** will be created in the **Node <node name> Upgrades** folder on the Logs page.

WEB SERVICES REST INTERFACE

The AWS Elemental Conductor File 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 AWS Elemental system. Explore features of the REST API below.

- REST Basics
    - HTTP Headers
    - API Versions
    - Simple Examples
    - Clean XML
    - Schema Definitions
    - Errors and Warnings
- Nodes
- Jobs
    - Example XML: Create a job from a profile
    - Example XML: Create a simple job using presets
    - Example XML: Create a more advanced job
    - Example XML: Overriding job profile output group settings
    - Example XML: Create a job from a profile with DRM overrides
    - Example XML: Advanced job profile overriding
- Job Profiles
- Job Watch Folders
- Presets
- Preset Categories
- Configuration
- Alerts and Messages
- Error Codes
- Warning Codes
- Audit Message Codes
- Query Parameters
- Pooled Licensing
- Authentication and REST
- Upgrading Deprecated XMLs

REST BASICS

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

HTTP HEADERS

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

API VERSIONS

When submitting REST requests manually or from within an automation system, it is recommended to use an API version prefix for all endpoints. The API version prefix allows you to specify which API version the server should use to interpret your data. For example, POST http://<server_ip>/api/v2.13.1/jobs will send a request to the /jobs endpoint, and the server will interpret the data as compatible with AWS Elemental API version 2.13.1. Although it is recommended that the API version prefix is included in all REST endpoints, omitting the prefix will assume the most current up-to-date API version: POST http://<server_ip>/api/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 AWS 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="2.13.1.xxxx" product="Elemental Conductor File">
  <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 describing the resource. The client application must set the HTTP "Content-Type" header to: Content-Type: application/xml.

To create a new job watch folder:

```
curl -H "Accept: application/xml" -H "Content-type: application/xml" \
     -d @filename http://<server_ip>/api/v2.13.1/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="2.13.1.xxxx" product="Elemental Conductor File">
  <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 AWS Elemental Conductor File 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 1. Simply make the regular GET request and add an extra parameter clean=true at the end.

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

This XML can be saved to a file and then POSTed back to the same server or another server to create an identical 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 -- from the Job Control page, click 'Job XML'.

SCHEMA DEFINITIONS

AWS Elemental products ship with XML schema definitions (XSDs) for the most common asset creation requests. These may be found in the /schema path as follows:

- /schema/Job.xsd - job creation schema
- /schema/JobProfile.xsd - job profile creation schema
- /schema/ServerPreset.xsd - preset creation schema

(Right/command-click >> Save As to download files.)

ERRORS AND WARNINGS

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

```
curl -H "Accept: application/xml" -H "Content-type: application/xml" \
     -d "<job_watch_folder></job_watch_folder>" http://<server_ip>/api/v2.13.1/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>
```

NODES

The following table describes the REST node control interface.

URL	METHOD	PARAMETERS	RETURNS	DESCRIPTION
/nodes	GET	Pagination parameters can be appended to the URL, ie /nodes?page=3&per_page=30	Node List Description	Retrieves a list of nodes connected to the system
/nodes/<id>	GET	A limit to the messages history (in hours) can be indicated by a parameter to the URL, ie /nodes/<id>?history=24. Default is 12 hours	Node Description	Retrieves information about a specific node, including node stats, a list of running Jobs, configuration parameters, and messages.
/nodes/<id>/add_to_cluster	POST			Reconfigures this node to be a member of the cluster
/nodes/<id>/remove_from_cluster	POST			Reconfigures this node to be a standalone AWS Elemental Conductor File node
/nodes/<id>/reconfigure	POST			Runs the configure script on this node with its current settings

URL	METHOD	PARAMETERS	RETURNS	DESCRIPTION
/nodes/<id>/reactivate	POST		Rejoin cluster	Change state of this node from "Waiting to Rejoin" to "Active". Node can be used for failover once active. Note that when Jobs are manually started on a node that is waiting, it will automatically become active.
/nodes/<id>/send_command	POST	<cmd> restart_elemental_se | stop_elemental_se | start_elemental_se </cmd>		Sends a command to the elemental_se service running on the Node in order to change it's state. Note that the service cannot be stopped, started, or restarted while Jobs are running.
/nodes/<id>/force_fail	POST			This will terminate any Jobs running on the node and cause it to be marked as failed. The node will rejoin the cluster after a minute. This can be useful for testing failure scenarios.
/nodes/<id>/config/network	GET		Network Settings	Retrieves information about the current network settings. <management_interface> is used to communicate with a Conductor unit in a clustered environment.
/nodes/<id>/config/mount_points	GET		Mount Point Settings	Retrieves information about the current mount point settings.
/nodes/<id>/config/firewall	GET		Firewall Settings	Retrieves information about the current firewall settings.
/nodes/<id>/config/advanced	GET		Advanced Settings	Retrieves information about the current advanced settings.

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>/outputs	GET		List of all outputs on the job	Retrieve list of all outputs on the job.
/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>
/jobs/<id>/cut_lists	GET		Job Cut Lists	Retrieves a list of job <id>'s cut lists. Cut lists are generated for each output where "Log Edit Points" is true
/jobs/<id>/cut_lists/<output_id>	GET		Job Output Cut List	Retrieves the cut list XML for output <output_id> in job <id>. <output_id> can be retrieved by using the above end point to get a list of all cut lists for job <id>

EXAMPLE XML: CREATE A JOB FROM A PROFILE

When specifying a profile, the id, name or permalink may be given. The system will search first for a profile id, then name and finally permalink until it finds a match. If no match is found, an error will be returned and the job will not be created.

```
<?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>
    <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>
        <spatial_aq>true</spatial_aq>
        <temporal_aq>false</temporal_aq>
        <flicker_aq>false</flicker_aq>
        <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>
        <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>
        <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>
        <spatial_aq>true</spatial_aq>
        <temporal_aq>false</temporal_aq>
        <flicker_aq>false</flicker_aq>
        <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>
        <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>
        <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>
      <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_frames_per_pes>16</audio_frames_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>
        <pat_interval>0</pat_interval>
        <pmt_interval>0</pmt_interval>
        <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>
      <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_frames_per_pes>16</audio_frames_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>
        <pat_interval>0</pat_interval>
        <pmt_interval>0</pmt_interval>
        <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 overridden 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="2.13.1.xxxx" product="Elemental Conductor File">
  <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>
        <spatial_aq>true</spatial_aq>
        <temporal_aq>false</temporal_aq>
        <flicker_aq>false</flicker_aq>
        <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"/>
        <scd>true</scd>
        <slices>1</slices>
        <slow_pal>false</slow_pal>
        <telecine>None</telecine>
        <profile>Main</profile>
        <rate_control_mode>CBR</rate_control_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>
```

```
          <speed nil="true"/>
        </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>
      <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 Job. 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="2.13.1.xxxx" product="Elemental Conductor File">
  <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>
        <spatial_aq>true</spatial_aq>
        <temporal_aq>false</temporal_aq>
        <flicker_aq>false</flicker_aq>
        <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"/>
        <scd>true</scd>
        <slices>1</slices>
        <slow_pal>false</slow_pal>
        <telecine>None</telecine>
        <profile>Main</profile>
        <rate_control_mode>CBR</rate_control_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>
          <speed nil="true"/>
        </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>
      <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>

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>

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
/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 Preset 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>

CONFIGURATION

Configuration provides information on overall cluster settings. The REST interface can only query information about the settings. Any settings updates must be made via the web interface.

URL	METHOD	RETURNS	DESCRIPTION
/config/failover	GET	Failover configuration	Auto rejoin: failed nodes should rejoin cluster when they come back online, or wait for operator intervention Failover threshold: number of seconds to wait until an unresponsive node is declared failed Heartbeat frequency: Number of seconds nodes wait between sending heartbeat messages to the Conductor system
/config/authentication	GET	Authentication Settings	Retrieves information about the current authentication settings.
/config/snmp	GET	SNMP Settings	Retrieves information about the current SNMP settings.
/config/advanced/conductor_file	GET	Cluster Sequencer Parameters	Retrieves cluster-wide settings for AWS Elemental Conductor File nodes.
/config/advanced/server	GET	Server Sequencer Parameters	Retrieves cluster-wide settings for AWS Elemental Server nodes.
/nodes/<id>/config/advanced	GET	Cluster Sequencer Parameters Server Sequencer Parameters	Updates node-specific settings for the given node.
/config/advanced/conductor_file	PUT	Cluster Sequencer Parameters	Updates cluster-wide settings for AWS Elemental Conductor File nodes.
/config/advanced/server	PUT	Sequencer Parameters	Updates cluster-wide settings for AWS Elemental Server nodes.
/nodes/<id>/config/advanced	POST	Cluster Sequencer Parameters Server Sequencer Parameters Stream File Sequencer Parameters	Updates node-specific settings for the given node. Please note that updating these settings will override any cluster-wide settings that were specified for this node.

ALERTS AND MESSAGES

The alerts API provides information about current alert conditions on the cluster. Messages provide an audit list of Jobs

URL	METHOD	PARAMETERS	RETURNS	DESCRIPTION
/alerts	GET	Pagination parameters, 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	Pagination parameters, 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 process input file *filename*.	Problem processing an input. The AWS Elemental Media Engine (EME) was unable to process an input.

CODE	ERROR MESSAGE	TROUBLESHOOTING
1020	No video stream in input file.	AWS Elemental products require at least one video stream in the input file. Audio only streams are not supported.
1021	No audio stream in input file.	When the audio encoder is configured for Dolby Digital Passthrough mode it is required that the input contains a Dolby Digital audio track.
1030	Unknown or unsupported video codec.	Check that this product supports the input source audio and video codec. See Supported 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, provide these details to AWS Elemental Support. Contact them through your company's Private Space on the AWS Elemental User Community.
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.
1056	Failed to open file	The specified file could not be opened. Check that the file or directory exists, and that the permissions allow the system to open the file.
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.
1075	Demuxer Parse Error	Demuxer could not recover from a problematic file. Details are given in the error message.
1076	Source Read Error	Source read error, possibly unexpected end of file. Details are given in the error message.
1077	Memory Allocation Error	Memory allocation failed. Details are given in the error message.
1080	MXF muxer error	An error in the MXF muxer has occurred. This could be from an invalid configuration (at job start) or from a runtime exception. Specific details are given in the error message.
1090	ESAM error	There was an error with ESAM. Details are given in the error message.
1091	Encryption error	There was an error encrypting the output. Details are given in the error message.
1100	Bonded encoding error	An error associated with bonded encoding has occurred.
1110	SDI driver version error	The SDI driver version is incorrect.
1111	SDI hardware firmware version error	The SDI hardware firmware version is incorrect.
1112	SDI Ingest communication error	There was an error communicating with another AWS Elemental process.
1999	*Critical EME (AWS Elemental Media Engine) error*	This code is returned for errors that require an AWS Elemental support technician to continue troubleshooting. Contact AWS Elemental Support through your company's Private Space on the AWS Elemental User Community.
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.
2056	Error logging in to access remote resource	Check that the credentials supplied are correct.
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 Error*	This code is returned for errors that require an AWS Elemental support technician to continue troubleshooting. Contact AWS Elemental Support through your company's Private Space on the AWS Elemental User Community.

WARNING CODES

The following is a list of 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 custom pre- or post- processing script executing before or after the Job returned a message.
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.
103010	Node marked as failed, possible network congestion preventing heartbeats. Consider increasing failover threshold	The node was marked as failed by the Conductor system, but it was still running. This can be caused by network congestion preventing the node's heartbeats from being delivered to the Conductor node in a timely manner.
103020	Node does not have capacity for Job *id*	A failure occurred and the Conductor system attempted to move the Job to this node, however the node did not have capacity to run the Job. The Conductor system will attempt to move the Job to another node.
103030	Job failed over from *hostname* to *hostname*	A failure occurred on a node and the Conductor system moved this Job to a backup node.

AUDIT CODES

The following list are common audit messages that the system may report.

CODE	AUDIT MESSAGE	DESCRIPTION
10	*Initial* timecode	HD-SDI inputs will report the initial timecode of the stream from the selected timecode source.
11	*Final* timecode	HD-SDI inputs will report the final timecodes of the stream from the selected timecode source.
12	*GOP* timecode	HD-SDI inputs will report the GOP timecodes of the stream from the selected timecode source.
13	Timecode Resync	The encoder will report when timecode is synced to timecode source.
30	Node Activated	An AWS Elemental Conductor File node has been activated in the cluster
31	Node Deactivated	An AWS Elemental Conductor File node has been shut down
32	Node added to cluster	A new AWS Elemental Conductor File node has been added to the cluster
33	Node removed from cluster	An AWS Elemental Conductor File node has been removed from the cluster
34	Node reconfiguring	An AWS Elemental Conductor File node is being reconfigured
35	Node shutdown	An AWS Elemental Conductor File node is in delayed shutdown
42	Encode Profile and Level	The encoder will report actual profile and level settings for each stream assembly.

QUERY PARAMETERS

The AWS Elemental Conductor File 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
```

Note: In many command shell environments, certain characters such as & may be interpreted as special characters so it is recommended to use a quoting mechanism specific to your shell. For example, issue the following command in the Bash shell using single or double quotes:

```
curl -H "Accept: application/xml" "http://<conductor_id>/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

POOLED LICENSING

To view a pooled license on a Conductor:

```
curl -H "Accept: application/xml" http://<conductor_id>/licenses
```

To view a pooled license being used by a worker node in the cluster through Conductor:

```
curl -H "Accept: application/xml" http://<conductor_id>/nodes/<worker_id>/licenses
```

To view a local license being used by a worker node at the worker node:

```
curl -H "Accept: application/xml" http://<worker_id>/licenses
```

AUTHENTICATION AND REST

When authentication is enabled on the AWS Elemental Conductor File 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, AWS Elemental recommends that this value should be ~30 seconds in the future.

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

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

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

For example, consider a GET request to https://<server_ip>/api/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'
```

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

AUTHCURL SCRIPTS

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

Using the same example from above, to send a GET request to '/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' https://<server_ip>/api/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 \
  -X [POST|PUT] \
  -H 'Accept: application/xml' -H 'Content-Type: application/xml' \
  -d @filename https://<server_ip>/api/v2.13.1/jobs
```

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

UPGRADING DEPRECATED XMLS

XMLs from older versions of AWS Elemental Conductor File may not be compatible with the current version of AWS Elemental Conductor File. If obsolete XMLs are used to create new Jobs, the system may return an error that recommends upgrading your XML using the AWS Elemental Conductor File XML update utility. This script can be found on your AWS Elemental Conductor File machine at:

```
/opt/elemental_se/web/public/helpers/elemental_xml_updater.rb
```

The utility can be used on the AWS Elemental Conductor File machine itself or any machine that has both Ruby and the libxml-ruby gem installed. For additional support, contact AWS Elemental Support through your company's Private Space on the AWS Elemental User Community.

JOB PARAMETERS

- General
- Location
- Input
  - Authentication Details
  - Video Selector
  - Audio Selector
  - Audio Selector Group
  - Caption Selector
    - Embedded Source Settings
    - File Source Settings
    - Teletext Source Settings
    - DVB Sub Source Settings
    - SCTE-27 Source Settings
  - Input Clipping
  - Image Inserter
  - Insertable Image
- Timecode Config
- Nielsen Configuration
- Processors
  - Notification
  - ESAM
  - Pre-Process
  - Post-Process
  - Image Inserter
  - Insertable Image
  - Avail Blanking
  - XDS Manipulation
  - Timed Metadata Insertion
  - ID3 Insertion
  - Progressive Reader Setting
- Output Group
  - File Group Settings
  - Apple Live Group Settings
  - Microsoft Smooth Streaming Group Settings
  - DASH ISO Group Settings
  - HDS Group Settings
  - Alternate Manifest Destination
  - Flash Access Settings
  - pHDS Settings

    - Verimatrix Settings
    - Secure Media Settings
    - Irdeto Settings
    - Conax Settings
    - Generic Keyprovider Settings
    - Self-Generated Settings
    - Piksel Settings
    - Inside Secure Settings
    - One Mainstream Settings
    - Cisco Settings
    - The Platform Settings
    - Seachange Settings
- Output
    - Apple Live Settings
    - MP4 Settings
    - F4V Settings
    - Mov Settings
    - UVU Settings
    - Raw Settings
    - M2TS Settings
        - DVB Network Information Table (NIT)
        - DVB Service Description Table (SDT)
        - DVB Time and Date Table (SDT)
        - Simulcrypt AES Settings
    - M3U8 Settings
- External Output
- Stream Assembly
    - Video Description
    - Rectangle
    - H.264 Settings
    - H.265 Settings
    - MPEG-2 Settings
    - ProRes Settings
    - Frame Capture Settings
    - Uncompressed Settings
    - Video Preprocessors
        - Color Corrector
        - Image Inserter
        - Insertable Image
        - Deinterlacer
        - Noise Reducer

        - Watermarking
        - Timecode Burn-in
    - Audio Description
    - AAC Settings
    - WAV Settings
    - AIFF 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
    - Burn-In Destination Settings
    - SCC Destination Settings
- Preset
    - Preset Category
    - Remix Settings Preset
- Job Profile
- Job Watch Folder
- Cluster Sequencer Config
- Format Identifier Parameters
- Scan Types

PARAMETERS

The following tables outline parameters that can be set for objects in AWS Elemental Conductor File. 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	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.
motion_image_inserter	Motion Image Inserter		Settings for the motion image inserter. When attached to the Job, inserts into the decoded input and appears in every output.
avail_blanking	Avail Blanking		Settings for ad avail blanking. Video can be blanked or overlaid with an image, and audio muted during SCTE-35 triggered ad avails.
esam	ESAM		Settings for Event Signaling And Messaging (ESAM).
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	Enables A/V sync trim audio.
nielsen_configuration	Nielsen Configuration		Nielsen configuration settings
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.
ad_avail_offset	integer	-1000 – 1000 (Default: **0**)	When specified, this offset (in milliseconds) is added to the input Ad Avail PTS time.

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 AWS Elemental Server system either on the local filesystem or through a SMB mount.
certificate_file	string		SSL Certificate file if using FTPS as input source.
authentication_details	Authentication Details		Specifies credentials for authenticated sources.

URI TYPES

The format for the URI may include a protocol. For some protocols, authentication with the remote server is supported and the username and password can be specified.

URIS AND PROTOCOLS

PROTOCOL	URI FORMAT	AUTHENTICATION
Local file	/data/server/folder/file.ext	None
CIFS or NFS mounted filesystem	/mnt/<mountpoint folder>/<filename>	See Mount Point Settings
HTTP	http://<web server>[:port]/path/file.ext	Basic and AWS (If authentication is required, enter the Access Key ID in the username field and the Secret Access Key in the password field.)
HTTPS	https://<web server>[:port]/path/file.ext	Basic and AWS (If authentication is required, enter the Access Key ID in the username field and the Secret Access Key in the password field.)
FTP	ftp://<ftp server>[:port]/path/file.ext	Basic
SFTP	sftp://<ftp server>[:port]/path/file.ext	Basic and SSH authentication; host key authentication (add /home/elemental/.ssh/id_rsa.pub to remote server's authorized key list).
SCP	scp://<remote server>[:port]/path/file.ext	Basic and SSH authentication; host key authentication (add /home/elemental/.ssh/id_rsa.pub to remote server's authorized key list).
Amazon S3	s3://<bucket>/<object>	Enter the Access Key ID in the username field. Enter the Secret Access Key in the password field. Use sse=true to enable S3 Server Side Encryption(SSE). Use rrs=true to enable Reduced Redundancy Storage (RRS). Default values for RRS and SSE are false. Example: s3://elemental.test/bucketname/encrypted?rrs=true&sse=true
Amazon S3SSL	s3ssl://<bucket>/<object>	Enter the Access Key ID in the username field. Enter the Secret Access Key in the password field. Use sse=true to enable S3 Server Side Encryption(SSE). Use rrs=true to enable Reduced Redundancy Storage (RRS). Default values for RRS and SSE are false. Example: s3ssl://elemental.test/bucketname/encrypted?rrs=true&sse=true
Aspera	aspera://<url>	Enter a username and password as specified by the server administrator.

PROTOCOLS FOR FILE INPUTS

Inputs that are file inputs can use the HTTP, HTTPS, FTP, SFTP, SCP, Amazon S3 and Aspera protocols. They can also be stored on local files or CIFS or NFS mounted filesystems.

HLS input can be handled as a file input by specifying HTTP or HTTPS as the protocol.

PROTOCOLS FOR OUTPUTS

For information on the protocols and formats for identifying the destination for outputs, see the individual parameters in each output group type.

INPUT

NAME	TYPE	RANGE	DESCRIPTION
file_input	Location		File to transcode. Must be accessible to the AWS Elemental Server node that is running the job through a local filesystem, an SMB mount, or a network connection. 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 within a multi-program transport stream. If the program doesn't exist, the first program within the transport stream will be selected by default. Use the preview button to populate the list of available programs.
filter_enable	string	Auto, **Disable**, or Force	Turns on the filter for this input. MPEG-2 inputs have the deblocking filter enabled by default. 1) Auto – filtering will be applied depending on input type/quality 2) Disable – no filtering will be applied to the input 3) Force – filtering will be applied regardless of input type
filter_strength	integer	1 – 5	Adjusts the magnitude of filtering from 1 to 5, with 1 being the nominal value.
denoise_selected	boolean	true or **false**	Allow the denoise filter when filtering.
deblock_selected	boolean	**true** or false	Allow the deblock filter when filtering.
no_psi	boolean	true or **false**	Only effective with Transport Stream inputs. Causes transport stream demux to scan all PIDs for audio and video rather than relying on PSI data.
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.
caption_selector	Caption Selector		Specifies a particular caption stream within an input source. An input may have multiple caption selectors.
supplemental_imp	Supplemental IMP		Contains path to supplemental Interoperable Master Format (IMF) Interoperable Master Packages (IMP's)
timecode_source	string	embedded, zerobased, systemclock, systemclock_local	Specifies the source of timecode associated with this input. Used for Input Clipping and input based Image Insertion. "Embedded" (embedded) will use the true timecode carried in the input. "Start at 0" (zerobased) associates 00:00:00:00 with the first frame of the input. "System Clock" (systemclock) uses UTC time. "Local System Clock" (systemclock_local) uses the UTC time adjusted for the timezone specified on the hardware unit where the event is running.
authentication_details	Authentication Details		Specifies credentials for authenticated sources.

AUTHENTICATION DETAILS

Credential storage for remote and local resources.

NAME	TYPE	RANGE	DESCRIPTION
authentication_type	string	basic, cross_account	Basic: Give AWS Elemental Server access to your S3 bucket by entering your AWS username and password. Not recommended. Cross-account dynamic authentication: Give AWS Elemental Server access to your S3 bucket by setting up cross-account access on your AWS account. Not enabled on ground appliances.
username	string		Required for basic authentication.
password	string		Required for basic authentication.
external_account_id	string		The ID of your AWS account that owns the specified S3 bucket. For example, 111122223333. Not enabled on ground appliances.
external_role_name	string		The name of the role you created in your AWS account. You created this role to give access to AWS Elemental Server. Not enabled on ground appliances.
external_id	string		The external ID you specified in the role. Not enabled on ground appliances.

VIDEO SELECTOR

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

NAME	TYPE	RANGE	DESCRIPTION
name	string		Selector name. This is used to attach selectors to input remix objects. This field is not saved, it is replaced with an id field once saved.
program_id	integer		Selects a specific program from within a multi-program transport stream. For Quadrant 4K inputs, this program carries the specified 1080p quadrant of the 2160p (4K) image. If the program doesn’t exist, the first program within the transport stream will be selected by default. Use the preview button to populate the list of available programs.
pid	decimal integer	> 0	Selects a specific PID from within a video source (e.g. 257 selects PID 0×101).
color_space	string	**follow**, rec_601, rec_709, hdr10, hlg_2020	Identifies the color space of the input. Typically set to Follow. Choose a specific color space only if the color space is (or is sometimes) missing from the input or if the color space is in the input but you know it is wrong. Also see force_color.
force_color	boolean		Applies only if color_space is a value other than Follow. This field controls how the value in the color_space field and values in the HDR Master Display Information fields will be used. Unchecked (false) means that when the input does include color space data, that data will be used, but when the input has no color space data, the value in color_space will be used. Choose false if your input has portions that are missing color space data, but when it does have color space data, that data is correct. Checked (true) means to always use the value in color_space. Choose true if your input usually has no color space data or might have unreliable color space data. In both cases if you set color_space to HDR10 and you don’t convert the color space in the output, then make sure you enter valid values in the HDR Master Display Information fields; these values will be used when the values in the input are not used. Make sure to obtain values used in the color grading process for the input; you cannot use the defaults or null values and expect to obtain valid color results.

AUDIO SELECTOR

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

NAME	TYPE	RANGE	DESCRIPTION
name	string		Selector name. This is used to attach selectors to input remix objects. This field is not saved, it is replaced with an id field once saved.
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.
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.
selector_type	string	pid, track, language_code	Specifies the type of the audio selector.
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).	Identify the input tracks to include in this selector by entering the 1-based track index. To combine several tracks, enter a comma-separated list, e.g. “1,2,3” for tracks 1-3.
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. Auto populated when Infer External Filename is checked
infer_external_filename	boolean	true or **false**	When checked, the external audio file name is inferred from the input (media) file name. If no language code is selected, audio file with same name as input media file are selected for transcoding. If language code is used, the audio file name with same name as input media file followed by _<3-char-language-code> is selected for the transcoding. The file naming convention is displayed in the External audio file name text box. If not checked then external audio file name has to be completely specified in the External Audio textbox. This feature is valid for local and S3/S3SSL inputs only.
strict_language_selection	boolean	true of **false**	When checked the transport stream demux strictly identifies audio streams by their language descriptor. If a PMT update occurs such that an audio stream matching the initially selected language is no longer present then mute will be encoded until the language returns. If not checked then on a PMT update the demux will choose another audio stream in the program with the same stream type if it can’t find one with the same language.
strict_pid_option	string		System will look for the specified PID before event start. If absent, "PID must be present in input" will prevent the event from starting, and "PID may be missing from input" allows the event to start with muted audio for the selector.
unwrap_smpte337	boolean	false or **true**	When checked, SMPTE-337-wrapped Dolby-E audio streams in the selector will be unwrapped and decoded by the Dolby decoder. If unchecked, such streams will be treated as raw PCM audio.

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.

NAME	TYPE	RANGE	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.

CAPTION SELECTOR

A Caption Selector is used to extract a specific type of caption data from a single input. When Caption Selectors are defined in the inputs, a Caption Description can then specify a *caption_source_name* in order to extract specific caption data across multiple inputs. Each input must contain the same number of Caption Selectors, and a special *Null* Caption Selector can be used to skip extraction from an input.

NAME	TYPE	RANGE	DESCRIPTION
source_type	string	**Embedded**, SCTE-20, SCC, Teletext, DVB-Sub, Ancillary, ARIB, TTML, SCTE-27, STL, SRT, SMI, IMSC, Null	Select the style of the captions. You must specify the style: the decoder cannot auto-detect the caption style. For more information, see "Setting Up Captions with AWS Elemental Server" in the public online documentation at https://docs.aws.amazon.com/elemental-server/latest/ug/setting-up-captions.html.
***source*_settings**	Source Settings	ancillary_source_settings, embedded_source_settings, file_source_settings, teletext_source_settings, dvb_sub_source_settings, scte27_source_settings, track_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.
order	integer	> 0	Required when an input has multiple caption selectors.

ANCILLARY SOURCE SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
source_ancillary_channel_number	integer	1 to 4	Specifies the 608 channel number in the ancillary data track from which to extract captions. Unused for passthrough.

EMBEDDED SOURCE SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
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. Unused for passthrough.
upconvert_608_to_708	boolean	true or **false**	If true, 608 data is both passed through via the "608 compatibility bytes" fields of the 708 wrapper as well as translated into 708. 708 data present in the source content will be discarded.
autodetect_scte20	boolean	true or **false**	Check to handle streams with intermittent and/or non-aligned SCTE-20 and Embedded captions.

FILE SOURCE SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
source_file	Location		External caption file used for loading captions. Accepted file extensions are scc, ttml, dfxp, stl, srt, and smi. Auto-populated when Infer External Filename is checked.
time_delta	integer		Specifies a time delta in seconds to offset the captions from the source file.

NAME	TYPE	RANGE	DESCRIPTION
infer_external_filename	boolean	true or **false**	When checked, the external caption file name is inferred from the input (media) file name. Caption file with the same name as input media file is selected for transcoding. The file naming convention is displayed in the Source File text box. If not checked then external caption file name has to be completely specified in the Source File textbox. This feature is valid for local and S3/S3SSL inputs only.
upconvert_608_to_708	boolean	true or **false**	If true, 608 data is both passed through via the “608 compatibility bytes” fields of the 708 wrapper as well as translated into 708. 708 data present in the source content will be discarded.

TELETEXT SOURCE SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
page_number	string	100-1FE, 200-2FE, 300-3FE, 400-4FE, 500-5FE, 600-6FE, 700-7FE, 800-8FE	Specifies the teletext page number within the data stream from which to extract captions. Must be a three-digit hexadecimal string. Unused for passthrough.
smpte_2031		true or **false**	When checked, OP-47 in SMPTE 2031 is used as input caption. When not checked, OP-47 is used. Only applies to SDI inputs.
embedded_caption_delay	integer	0 to 3000	Specifies a time in milliseconds to delay the captions from the source video. Only applies to SDI inputs.

DVB SUB SOURCE SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
pid	decimal integer	> 0	When using DVB-Sub with Burn-In or SMPTE-TT, use this PID for the source content. Unused for DVB-Sub passthrough. All DVB-Sub content is passed through, regardless of selectors.

SCTE-27 SOURCE SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
pid	decimal integer	> 0	The specific language to extract from the source: Specify PID and Language: Extracts captions from that PID; the language is “informational”. Specify PID and omit Language: Extracts the specified PID. Omit PID and specify Language: Extracts the specified language, whichever PID that happens to be. Omit PID and omit Language: Valid only if source is DVB-Sub that is being passed through; all languages will be passed through.

TRACK SOURCE SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
track_number	integer	Positive integers	Specifies the captions track within an IMF composition playlist. 1 indicates the first track.

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 of the final frame of the clip. This frame is included in the clip. 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.

IMAGE INSERTER

The image inserter overlays a 32-bit Windows BMP, PNG or TGA 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*.

NAME	TYPE	RANGE	DESCRIPTION
enable_rest	boolean	true or **false**	Indicates that REST commands will be used to send image insertion commands. If used, no other fields are needed.
insertable_image	Insertable Image		Image to insert. Must be 32 bit windows BMP, PNG, or TGA file. Must not be larger than the output frames.

INSERTABLE IMAGE

NAME	TYPE	RANGE	DESCRIPTION
image_inserter_input	Location		Image to insert. Must be 32 bit windows BMP, PNG or TGA. Must not be larger than the output frames.
layer	integer	0 – 99	The Z order of the inserted image. Images with higher values of layer will be inserted on top of images with lower values of layer.
image_x	integer		Placement of image on the horizontal axis in pixels. 0 is the left edge of the frame. Required for BMP, PNG and TGA input.
image_y	integer		Placement of image on the vertical axis in pixels. 0 is the top edge of the frame. Required for BMP, PNG and TGA input.
opacity	integer	0 – 100 (Default: **50**)	Opacity of image. 0 is transparent. 100 is fully opaque. Required for BMP, PNG and TGA input.
width	integer		The width of the image when inserted in the video. Leave blank to use the native width of the image.
height	integer		The height of the image when inserted in the video. Leave blank to use the native height of the image.
start_time	string		The start time for the image. Must be in timecode (HH:MM:SS:FF) format.
duration	integer		The time in milliseconds for the image to remain in the video.
fade_in	integer		The time in milliseconds for the image to fade in.
fade_out	integer		The time in milliseconds for the image to fade out.

TIMECODE CONFIG

NAME	TYPE	RANGE	DESCRIPTION
source	string	**embedded**, zerobased, specifiedstart	Identifies the source of the time that will be associated with the event. Time in the event runs on a clock (not on a timer). Regardless of the source, the time will be in 24-hour format hh:mm:ss:ff and will roll over at midnight. **Embedded (embedded)**: Use the timecode in the source video. If no embedded timecode is detected in the source, the system falls back to using "Start at 0" (zerobased). **Start at 0 (zerobased)**: The time of the first frame of the event will be 00:00:00:00. **Specified Start (specifiedstart)**: The time of the first frame of the event will be the time specified in the start parameter.
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.
require_initial_timecode	boolean		Event won't start if timecode source is lost at the start time.
override_timecode_date	boolean	**true** or false	When checked, use timestamp_offset to indicate the desired date (as well as the time) in those outputs, such as HLS, that support program dates (datestamps). When unchecked, use the encode date as the program date.
sync_threshold	integer	1 – 1000000 or nil	Threshold in frames beyond which output timecode is resynchronized to the input timecode. Discrepancies below this threshold are permitted to avoid unnecessary discontinuities in the output timecode. No timecode sync when this is not specified. If jam sync is also defined, discrepancies beyond this threshold would not cause resync but only trigger alerts.
timestamp_offset	date		Date to use when inserting date in outputs that support program dates. Only applies if insert_program_date_time is true and override_timecode_date is true.

NIELSEN CONFIGURATION

NAME	TYPE	RANGE	DESCRIPTION
breakout_code	integer	0, 3, 7, 9	Nielsen measurement system breakout code. Contact Nielsen for more information on usage.
distributor_id	string		Distribution ID. Enter the Distribution ID assigned to your organization by Nielsen

PROCESSORS

ESAM

NAME	TYPE	RANGE	DESCRIPTION
scc_uri	Location		URL of the Signal Conditioner endpoint. if used, should contain a URL. Used to process signal conditioning information, which is when and where to insert IDR's.
alternate_scc_uri	Location		URL of an Alternate Signal Conditioner endpoint. Only used if the primary Signal Conditioner endpoint is not available.

NAME	TYPE	RANGE	DESCRIPTION
mcc_uri	Location		URL of the Manifest Conditioner endpoint. if used, should contain a URL. Used to process manifest conditioning information, which is how to manipulate the manifest (only applies to HLS outputs). If empty, no manifest manipulation is performed.
alternate_mcc_uri	Location		URL of an Alternate Manifest Conditioner endpoint. Only used if the primary Manifest Conditioner endpoint is not available.
acquisition_point_id	string		A system-wide unique string identifying the transcoder/packager at a specific site on a specific channel/network feed.
asset_uri_id	string		An identifier of the asset being processed that is passed to the Signal Conditioner as part of the UriProcessingRequest message.
scc_xml	string		Specifies an ESAM SignalProcessingNotification XML document as per OC-SP-ESAM-API-I03-131025. If this SignalProcessingNotification XML is specified at job start time, then the transcoder will not contact the POIS system and use the Signal Processing instructions in the supplied message. When supplied over the REST API the XML Document must be wrapped in CDATA Section.
mcc_xml	string		Specifies an ESAM ManifestConfirmConditionNotification XML document as per OC-SP-ESAM-API-I03-131025. If this ManifestConfirmConditionNotification XML is specified at job start time, then the transcoder will not contact the POIS system and use the Manifest Conditioning instructions in the supplied message. When supplied over the REST API the XML Document must be wrapped in CDATA Section.
response_signal_preroll	integer	0 – 30000	Specifies the stream distance between the placement of POIS supplied SCTE 35 messages and the splice points that they refer to. If there is insufficient notification time to honor the entire pre-roll, then the SCTE 35 message will be placed immediately.
enable_nptpoint_feedback	boolean		When checked, generates an XML file esam_signal_points_feedback_job<ID>.xml in the job's log directory to log the original NPT values and processed NPT values of all the signal points.

NOTIFICATION

Notification objects allow AWS 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
script	Location		Script to run before Job starts.
progressive_reader_setting	Progressive Reader Setting		File handling for this job.

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. Only available for S3 sources that are using the 'S3://' or 'S3SSL://' protocols.
delete_source_dir	boolean	true or **false**	Deletes input source directory and all files and subdirectories when deleting source. Note this will affect preroll or postroll source files if they are in the source directory. Not available for S3 input sources.
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.

XDS MANIPULATION

Indicates XDS Manipulation will be enabled.

NAME	TYPE	RANGE	DESCRIPTION
enabled	boolean	true or **false**	
vchip_action	string	**Passthrough**, Rewrite, Insert, Strip	Content Advisory action.
vchip_byte1	string		Character 1 value of the Content Advisory, as per CEA-608 Line 2.1 Data Services, section 9.5.1.5 Type=0×05. Entry should be an integer byte in hexadecimal.
vchip_byte2	string		Character 2 value of the Content Advisory, as per CEA-608 Line 2.1 Data Services, section 9.5.1.5 Type=0×05. Entry should be an integer byte in hexadecimal.
copy_protection_action	string	**Passthrough**, Rewrite, Insert, Strip	Copy and Redistribution Control Packet action.
copy_protection_byte1	string		Character 1 value of the CGMS-A services and Analog Protection Services (APS), as per CEA-608 Line 2.1 Data Services, section 9.5.1.8 Type=0×05. Entry should be an integer byte in hexadecimal.
copy_protection_byte2	string		Character 2 value of the CGMS-A services and Analog Protection Services (APS), as per CEA-608 Line 2.1 Data Services, section 9.5.1.8 Type=0×05. Entry should be an integer byte in hexadecimal.

TIMED METADATA INSERTION

Provides the ability to inject ID3 tags to output at arbitrary points, defined via a sequence of ID3 tags and timecodes.

NAME	TYPE	RANGE	DESCRIPTION
enable	boolean	true or **false**	Must be specified as true when included in Job XML.
id3_insertion	ID3 Insertion		ID3 metadata to insert. Composed of a timecode and the ID3 tag.

ID3 INSERTION

A single instance of ID3 tag to be inserted at the specified timecode.

NAME	TYPE	RANGE	DESCRIPTION
timecode	string		Must be in NN:NN:NN:NN or NN:NN:NN;NN format.
id3	string		Base64-encoded ID3 tag to be inserted at the provided timecode.

IMAGE INSERTER

The image inserter overlays a 32-bit Windows BMP, PNG or TGA 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*.

NAME	TYPE	RANGE	DESCRIPTION
enable_rest	boolean	true or **false**	Indicates that REST commands will be used to send image insertion commands. If used, no other fields are needed.
insertable_image	Insertable Image		Image to insert. Must be 32 bit windows BMP, PNG, or TGA file. Must not be larger than the output frames.

INSERTABLE IMAGE

NAME	TYPE	RANGE	DESCRIPTION
image_inserter_input	Location		Image to insert. Must be 32 bit windows BMP, PNG or TGA. Must not be larger than the output frames.
layer	integer	0 – 99	The Z order of the inserted image. Images with higher values of layer will be inserted on top of images with lower values of layer.
image_x	integer		Placement of image on the horizontal axis in pixels. 0 is the left edge of the frame. Required for BMP, PNG and TGA input.
image_y	integer		Placement of image on the vertical axis in pixels. 0 is the top edge of the frame. Required for BMP, PNG and TGA input.
opacity	integer	0 – 100 (Default: **50**)	Opacity of image. 0 is transparent. 100 is fully opaque. Required for BMP, PNG and TGA input.
width	integer		The width of the image when inserted in the video. Leave blank to use the native width of the image.
height	integer		The height of the image when inserted in the video. Leave blank to use the native height of the image.
start_time	string		The start time for the image. Must be in timecode (HH:MM:SS:FF) format.
duration	integer		The time in milliseconds for the image to remain in the video.
fade_in	integer		The time in milliseconds for the image to fade in.
fade_out	integer		The time in milliseconds for the image to fade out.

MOTION IMAGE INSERTER

A Motion Image Inserter implements a processing stage that consumes pictures from a FIFO and produces pictures into a downstream FIFO after optionally merging the picture with multiple graphics layers. When using REST, you can omit all fields.

NAME	TYPE	RANGE	DESCRIPTION
insertion_mode	string	mov, png, swf	MOV, PNG or SWF.
motion_image_inserter_input	Motion Image Inserter Input		Motion image / image sequence to insert. Must be MOV, PNG or SWF.

NAME	TYPE	RANGE	DESCRIPTION
image_x	integer		X offset to place image, from top-left corner of video. Incompatible with full_frame, required otherwise.
image_y	integer		Y offset to place image, from top-left corner of video. Incompatible with full_frame, required otherwise.
enable_rest	boolean	true or **false**	Indicates that REST commands will be used to send image insertion commands. If used, no other fields are needed. Cannot be set via REST.
loop_input	boolean	true or **false**	Repeat playback of input or play only once.
active	boolean	true or **false**	
full_frame	boolean	true or **false**	Expand to fit frame. Preserves aspect ratio of images.
framerate_numerator	integer		Required with PNG mode. Framerate ratio must be between 1 and 120.
framerate_denominator	integer		Required with PNG mode. Framerate ratio must be between 1 and 120.
action_time	string		May be in timecode (HH:MM:SS:FF) or ISO 8601 UTC Timestamp (20150101T120000.1283) format, no dashes or colons. Leave out or leave empty for immediate activation.
duration	integer		The time in milliseconds for the image to remain in the video. If left blank, the duration of the file is used.
swf_arguments	string		SWF arguments in simple JSON name/value format

PROGRESSIVE READER SETTING

NAME	TYPE	RANGE	DESCRIPTION
progressive_reader_setting	string	use_global_setting, enable, disable_and_copy_all, disable	Input files from remote resources (s3, s3ssl, http, https) will be fetched according to setting: - enable: fetch dynamically during transcode - disable: prefetch before transcode, - disable_and_copy_all: prefetch from s3, s3ssl, http, https and external drives, - use_global_setting: will follow the progressive reader setting (in Advanced Settings). This option is only available in Pre Processing. This setting is not available in cloud deployments.

OUTPUT GROUP

NAME	TYPE	RANGE	DESCRIPTION
type	string	file_group_settings, apple_live_group_settings, hds_group_settings, ms_smooth_group_settings, dash_iso_group_settings, cmaf_group_settings	Output group settings type – defines the type of this output group.
name	string		
custom_name	string		Custom group name to be defined by user. Only letters, numbers and the underscore character allowed; only 12 characters allowed.
order	integer	> 0	Required for multiple output groups. Specifies the order the output groups should be listed in.
*output_group*_settings	Group Settings	file_group_settings, apple_live_group_settings, dash_iso_group_settings, hds_group_settings, ms_smooth_group_settings, cmaf_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. See Uri Types for supported protocols.
rollover_interval	integer	0 – 31557600	Rollover interval. AWS Elemental Live only; nil on AWS Elemental Server.

APPLE LIVE GROUP SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
destination	Location		A directory or HTTP destination for the HLS segments, manifest files, and encryption keys (if enabled). To enable HTTP Push, also select an ‘HTTP Push Dialect’ option below.
base_url_content	string		A partial URI prefix that will be prepended to each output in the media .m3u8 file. Can be used if base manifest is delivered from a different URL than the main .m3u8 file.
base_url_manifest	string		A partial URI prefix that will be prepended to each output in the media .m3u8 file. Can be used if base manifest is delivered from a different URL than the main .m3u8 file.
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.
min_segment_length	integer	0 – segment_length (Default: **0**)	When set, Minimum Segment Size is enforced by looking ahead and back within the specified range for a nearby avail and extending the segment size if needed.
emit_single_file	boolean	true or **false**	Emits program as a single media resource (.ts) file, uses #EXT-X-BYTERANGE tags to index segment for playback.
floating_point_manifest	boolean	**true** or false	Indicates whether the output manifest should use floating point values for segment duration.
include_resolution	boolean	**true** or false	Include RESOLUTION attribute for video in EXT-X-STREAM-INF tag of variant manifest.
compress_manifests	boolean	true or **false**	Compresses HLS playlist with gzip when enabled.
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 value is calculated as follows: either the program date and time are initialized using the input timecode source, or the time is initialized using the input timecode source and the date is initialized using the timestamp_offset.
timed_metadata_id3_period	integer		Timed Metadata interval in seconds.
timed_metadata_id3_frame	string	None, PRIV, TDRL	Indicates ID3 frame that has the timecode.
program_date_time_period	integer	0 – 3600 seconds (one hour)	Period of insertion of EXT-X-PROGRAM-DATE-TIME entry, in seconds.
cdn	string	None, Basic_PUT, Akamai, WebDAV	Type of HTTP communication to use for pushing to origin server: 1) None – not allowed for HTTP destinations 2) Basic_PUT – HTTP PUT with non-chunked encoding. No deletion of old files and no creation of folders. 3) WebDAV – HTTP PUT, PROPFIND, MKCOL, DELETE. Chunked Encoding. Creates folders as needed and deletes old files. 4) Akamai – compatibility with Akamai CDN inputs.
connection_retry_interval	integer	>= 0 (Default: **2**)	Number of seconds to wait before retrying connection to the CDN if the connection is lost.

NAME	TYPE	RANGE	DESCRIPTION
generate_meta_file	boolean	**true** or false	Generates the .m3u8 playlist file for this HLS output group. Unchecking this option will output segments without the .m3u8 file.
vod_mode	boolean	true or **false**	Keeps and indexes all segments starting with the first segment. Players will start playback at the beginning as they would with VOD.
num_retries	integer	>= 0 (Default: **10**)	Number of retry attempts that will be made before the Job is put into an error state.
filecache_duration	integer	0 – 600	Size in seconds of file cache for streaming outputs.
alternate_manifest_destination	Alternate Manifest Destination		The set-level and stream-level manifests will be additionally pushed to this location. Filenames will be identical to those pushed to the primary location. Each file will be pushed immediately after a successful push to the primary location. This destination type (Local, HTTP, etc.) must match the primary location. Do not specify a destination basename. The location must end with a trailing forward slash.
encryption_type	string	AES-128, SAMPLE-AES, pHLS, Flash_Access	Encrypts the segments with the given encryption scheme. Leave blank to disable. Selecting 'Disabled' in the web interface also disables encryption.
caption_language_setting	string	insert, omit, none	Applies only to 608 Embedded output captions. Insert: Include CLOSED-CAPTIONS lines in the manifest. Specify at least one language in the CC1 Language Code field. One CLOSED-CAPTION line is added for each Language Code you specify. Make sure to specify the languages in the order in which they appear in the original source (if the source is embedded format) or the order of the caption selectors (if the source is other than embedded). Otherwise, languages in the manifest will not match up properly with the output captions. None: Include CLOSED-CAPTIONS=NONE line in the manifest. Omit: Omit any CLOSED-CAPTIONS line from the manifest.
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.
show_iv	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, IV is listed in the manifest. If disabled, IV is not listed.
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 represented 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, static_key_settings, verimatrix_settings, secure_media_settings, irdeto_settings, conax_settings, generic_keyprovider_settings, piksel_settings, inside_secure_settings, one_mainstream_settings, cisco_settings, the_platform_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 delimited 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.

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. Value is accepted only when no key provider (self-generated) is specified.
ad_markers	string	adobe, elemental, elemental-scte35	Choose one or more ad marker types to pass SCTE35 signals through to this group of Apple HLS outputs.
disable_cache	boolean	true or **false**	When true, sets #EXT-X-ALLOW-CACHE:no tag, which prevents client from saving media segments for later replay.
use_pantos_7_codecs	boolean	true or **false**	When true, uses RFC-6381 instead of the default RFC-4281 during m3u8 playlist generation.
policy_file	Location		A file which contains the rules and restrictions that determine how, when, and where protected content can be viewed by consumers.
swf_identifiers_file	Location		Specifies a file of hashes of SWF players that are approved players for this content. Use the Adobe Media Server whitelist tool to generate these files.

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**	Number of bits to use in the IV.
initial_iv	integer	Default: **1**	Initial value of IV.
key_id	string	GUID	Specifies a key ID to use for Playready DRM, must be a valid GUID.
key_seed	string	base64 encoded	Contains a base64-encoded key seed. Only required if content_key is not specified.
content_key	string	base64 encoded	Contains a base64-encoded content key. If exists, key_seed is not required and ignored.
license_url	string		Contains the URL for the license acquisition Web service.
ui_license_url	string		Contains the URL for a non-silent license acquisition Web page.
custom_attributes	string		The content author can add arbitrary custom attributes inside this element. Microsoft code does not act on any data contained inside this element.
key_provider_settings	Key Provider Settings	irdeto_settings, seachange_settings, conax_settings, piksel_settings, inside_secure_settings	Key Provider-specific settings.
manifest_encoding	string	**utf-8** or utf-16	Text encoding to use for server and client manifests
collapse_identical_audio_streams	boolean		When checked, audio streams with the same settings are removed from all but one of the video streams.

DASH ISO GROUP SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
destination	Location		A directory or HTTP destination for DASH streaming. When Media Content Destination is specified, only MPD and initialization segment are sent here.
emit_single_file	boolean	**true** or false	When set to true, a single output file is generated, which is internally segmented using the Fragment Length and Segment Length. When set to false, separate segment files will be created.
fragment_length	integer	>= 1 (Default: **2**)	Length of fragments to generate (in seconds). Fragment length must be compatible with GOP size and Framerate. Note that fragments will end on the next keyframe after this number of seconds, so actual fragment length may be longer. When Emit Single File is checked, the fragmentation is internal to a single output file and it does not cause the creation of many output files as in other output types.
segment_length	integer	>= 1 (Default: **30**)	Length of mpd 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. When Emit Single File is checked, the segmentation is internal to a single output file and it does not cause the creation of many output files as in other output types.
base_url	string		A partial URI prefix that will be put in the manifest (.mpd) file at the top level BaseURL element. Can be used if streams are delivered from a different URL than the manifest file.
hbbtv_enabled	boolean	true or **false**	Supports HbbTV specification version 1.5
min_buffer_time	integer	>= 0	Minimum time of initially buffered media that is needed to ensure smooth playout.
drm_system	string	**nil** or widevine	Specifies DRM system used for DASH outputs. None by default.
key_provider_settings	Key Provider Settings	piksel_settings	Key Provider-specific settings.

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.
format_type	string	**fragmented**, segmented, url_defined	Determines the file type emitted from the HDS packager. Type 1 emits f4x segment files. Type 2 emits fragment files in the form of SegXXX-FragYYY. Segment files are typically used for VOD applications and fragment files are typically used for live streaming applications. “URL Defined” determines the type emitted based on the Destination URL, where local destinations emit Type 1 and http:// destinations emit Type 2; this matches the historical behavior of AWS Elemental Live.
fragment_length	integer	>= 1 (Default: **3**)	Length of fragments to generate (in seconds). Fragment length must be compatible with GOP size and Framerate.
min_fragment_length	integer	0 – fragment_length (Default: **0**)	When set, Minimum Fragment Length is enforced by looking ahead and back within the specified range for a nearby avail and extending the fragment length if needed.
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.
use_absolute_time	boolean	**true** or false	If true, use the time since start of the UNIX Epoch as the timestamp, otherwise, use the time since the start of the encode.
encryption_type	string	**nil**, flash_access, pHDS	Encrypts the fragments with the given encryption scheme when using Adobe HDS DRM.

NAME	TYPE	RANGE	DESCRIPTION
encryption_settings	Encryption Settings	flash_access_settings, phds_settings	Encryption-specific settings.
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.
ad_signaling	string	onCuePointSCTE35, PrimetimeDPISimple, PrimetimeDPISCTE35	Controls the method of signaling ads in HDS streams. onCuePoint SCTE35 takes all incoming SCTE104 or SCTE35 messages and translates them to a similar form for insertion as timed onCuePoint key/value pairs. Enough data is carried to fully reconstruct the original SCTE35 command. Primetime DPI Simple writes <cue> tags into the manifest to indicate times and durations of ad placements. Primetime DPI SCTE35 also writes into the manifest, but adds a Base64 encoded copy of the original SCTE35 message. These options are not exclusive, and some or all may be selected.
bootstrap_location	string	embedded, external	HDS Bootstrap files control can be emitted as a separate file (with a *.bootstrap extension), or embedded as data in HDS manifest files of version 3.0 and above. The control selects the output type. VOD output requires embedded bootstraps.

ALTERNATE MANIFEST DESTINATION

NAME	TYPE	RANGE	DESCRIPTION
destination	Location		The set-level and stream-level manifests will be additionally pushed to this location. Filenames will be identical to those pushed to the primary location. Each file will be pushed immediately after a successful push to the primary location. Do not specify a destination basename. The location must end with a trailing forward slash.
cdn	string	None, Basic_PUT, Akamai, WebDAV	Type of HTTP communication to use for pushing to origin server: 1) None – not allowed for HTTP destinations 2) Basic_PUT – HTTP PUT with non-chunked encoding. No deletion of old files and no creation of folders. 3) WebDAV – HTTP PUT, PROPFIND, MKCOL, DELETE. Chunked Encoding. Creates folders as needed and deletes old files. 4) Akamai – compatibility with Akamai CDN inputs.
connection_retry_interval	integer	>= 0 (Default: **2**)	Number of seconds to wait before retrying connection to the CDN if the connection is lost.
num_retries	integer	>= 0 (Default: **10**)	

FLASH ACCESS SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
generate_cek	boolean	**true** or false	When checked, the Encryption Key and Content ID are combined to generate a unique Content Encryption Key (CEK). When unchecked, the Encryption Key is used directly as the CEK. When checked, the Encryption Key can be a file of arbitrary length. When unchecked, the Encryption Key file must be 16 bytes (128 bits) long.
video_encrypt_level	string	**low**, medium, high	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.

NAME	TYPE	RANGE	DESCRIPTION
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.
swf_identifiers_file	Location		Specifies a file of hashes of SWF players that are approved players for this content. Use the Adobe Media Server whitelist tool to generate these files.
key_server_certificate	Location		Certificate required to support an embedded (non-chained) license with Remote Key Delivery.
recipient_certificate	Location		A certificate which uniquely identifies the recipient machine and client instance.

PHDS SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
encrypt_video	boolean	**true** or false	If true, video data is encrypted by the pHDS encryption, otherwise it is sent in the clear.
encrypt_audio	boolean	**true** or false	If true, audio data is encrypted by the pHDS encryption, otherwise it is sent in the clear.
encrypt_script	boolean	**true** or false	If true, script (metadata) data is encrypted by the pHDS encryption, otherwise it is sent in the clear.
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.
encryption_seed	string		Specify some text that will be hashed to create the 128 bit Content Encryption Key for this output. Synchronizing this text in multiple Events will cause the same key to be generated in both Events. This is recommended practice for failover scenarios.
policy_file	Location		A file which contains the rules and restrictions that determine how, when, and where protected content can be viewed by consumers.
swf_identifiers_file	Location		Specifies a file of hashes of SWF players that are approved players for this content. Use the Adobe Media Server whitelist tool to generate these files.

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.
use_rotating_keys	boolean		
program_identifier	string		
program_identifier	Location		
program_identifier	Location		
program_identifier	Location		

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.

STATIC KEY SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
static_key_value	string		Static Key value.
keyprovider_server	Location		The URL of the license server used for protecting content.

SELF-GENERATED SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
common_key	boolean	true or **false**	When enabled, generates the same key for each output within this output group.

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.

PIKSEL SETTINGS

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

INSIDE SECURE SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
key_provisioning_server	Location		Specifies the Location of the Inside Secure (Authentec) Key Provisioning server.
la_server	Location		Specifies the location of the License Acquisition server.

ONE MAINSTREAM SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
channel_secret	string		String used to sign encryption requests. Provided by 1Mainstream.
content_id	string		Video id.
channel_code	string		Channel code.
base_url	Location		The URL of the license server used for protecting content.

CISCO SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
key_id	string		Expected to be in GUID format.
key_seed	string		Expected to be base64 encoded.
la_url	Location		The URL of the license server used for protecting content.

THE PLATFORM SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
server	string		The content key ID.
key_value	string		The value of the AES-128 encryption key. Specified in hex with no 0x prefix.
target_client	enum	**inside_secure**, irdeto, microsoft	Target client.
la_url	Location		The license acquisition URL.

SEACHANGE SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
License Acquisition URL	Location		The URL of the license server used for protecting content.
Key Server	Location		
Client Certificate	Location		Provide the path and file name for the client certificate.
program_identifier	string		

OUTPUT

An output object describes the settings for a single output file or stream in an output group.

NAME	TYPE	RANGE	DESCRIPTION
description	string		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.
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.
name_modifier	string		String concatenated to the end of the destination filename. Only applicable to Archive, Apple HLS, MS Smooth, DASH ISO and Adobe HDS outputs. Required for multiple outputs of the same type. Accepts Format Identifiers. For DASH ISO outputs, if the format identifiers $Number$ or $Time$ are used in one output, they must be used across all outputs within the group.
extension	string		Output file extension. Applies to archive outputs. If empty, this will be auto-selected from the container type.
container	enum		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.
output_settings	Output Settings	apple_live_settings	Specific settings for this type of output.
*container*_settings	Container Settings	mov_settings, uvu_settings, m2ts_settings, raw_settings, mp4_settings, f4v_settings	Container specific settings. Note: replace *container* with the container you are using in the XML tag (e.g. <mov_settings>).
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.
insert_scte35_esam	boolean	true or **false**	If true, update any SCTE-35 signals from ESAM POIS to this output. Only available for m2ts 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.
insert_timed_metadata	boolean	true or **false**	If true, inserts ID3 timed metadata from the timed_metadata REST command into this output. Only available for certain containers.
start_paused	boolean	true or **false**	If true, output will start in the paused state.
log_edit_points	boolean	true or **false**	Generates an XML file in the log directory with initial timecode, timecode of input switches, and final timecode. This can be used to for later editing of this output.

NAME	TYPE	RANGE	DESCRIPTION
write_hvc1_for_h265	boolean		If true, output that is H.265 will be marked as HVC1 and adhere to the ISO-IECJTC1-SC29_N13798_Text_ISOIEC_FDIS_14496-15_3rd_E spec which states that parameter set NAL units will be stored in the sample headers but not in the samples directly. If this is unchecked **false**, then H.265 will be marked as HEV1 and parameter set NAL units will be written into the samples.

APPLE LIVE SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
segment_modifier	string		String concatenated to end of segment filenames. Accepts Format Identifiers.
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.}.
audio_track_type	string	**alternate_audio_auto_select_default**, alternate_audio_auto_select, alternate_audio_not_auto_select,audio_only_variant_stream	Four types of audio-only tracks are supported: Audio-Only Variant Stream The client can play back this audio-only stream instead of video in low-bandwidth scenarios. Represented as an EXT-X-STREAM-INF in the HLS manifest. Alternate Audio, Auto Select, Default Alternate rendition that the client should try to play back by default. Represented as an EXT-X-MEDIA in the HLS manifest with DEFAULT=YES, AUTOSELECT=YES Alternate Audio, Auto Select, Not Default Alternate rendition that the client may try to play back by default. Represented as an EXT-X-MEDIA in the HLS manifest with DEFAULT=NO, AUTOSELECT=YES Alternate Audio, not Auto Select Alternate rendition that the client will not try to play back by default. Represented as an EXT-X-MEDIA in the HLS manifest with DEFAULT=NO, AUTOSELECT=NO
iframe_only_manifests	boolean	true or **false**	Adds I-Frame Only Manifest in addition to the HLS manifest
audio_rendition_sets	string		List all the audio groups that are used with the video output stream. Input all the audio GROUP-IDs that are associated to the video, separate by ‘,’.
audio_group_id	string		Specifies the group to which the audio Rendition belongs.

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.
mp4_major_brand	string		Overrides the “Major Brand” field in the output file. Usually not necessary to specify.

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

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.
include_cslg	boolean	**true** or false	When checked file composition times will start at zero, composition times in the ‘ctts’ (composition time to sample) box for B-frames will be negative, and a ‘cslg’ (composition shift least greatest) box will be included per 14496-1 amendment 1. This improves compatibility with Apple players and tools.
growing_reference	boolean	**true** or false	If checked the Quicktime external reference file will be written out every 30 seconds, with pointers to the media file content up to that moment. If unchecked, no reference file will be written until the Event completes. Enabling this feature allows loading the content into an NLE program and to begin using it before the Event completes. Only valid with External reference.
write_xdcam	boolean	**false** or true	Enable XDCAM for Apple editors and players; Uncheck this box to support other players.
omneon_padding	boolean	**true** or false	Insert Omneon-compatible padding

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
encrypted	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.
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.

NAME	TYPE	RANGE	DESCRIPTION
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 media'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)
```

RAW SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
smpte_436_anc_passthrough	boolean	**false** or true	If true, passes through ancillary data from the source to SMPTE 436-M data in the output.

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.
program_num	integer	0 – 65535 (Default **1**)	The value of the program number field in the Program Map Table.
pat_interval	integer	0, 10 – 1000 (Default: **100**)	The number of milliseconds between instances of this table in the output transport stream.
pmt_interval	integer	0, 10 – 1000 (Default: **100**)	The number of milliseconds between instances of this table in the output transport stream.
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 – 500	Maximum time in milliseconds between Program Clock References (PCRs) inserted into the transport stream.
transport_stream_id	integer	0 – 65535	The value of the transport stream ID field in the Program Map Table.
use_buffer_model	boolean	**true** or false	Use multiplex buffer model for accurate interleaving. Disabling use_buffer_model can lead to lower latency, but low-memory devices may not be able to play back the stream without interruptions.
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.
use_atsc_stream_type	boolean	true or **false**	When checked, uses stream type = 0×81 for AC3 and stream type = 0×87 for EAC3. Otherwise uses stream type = 0×06.
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_frames_per_pes	integer	>= 0 (Default: **2**)	The number of audio frames to insert for each PES packet.
audio_buffer_compliant	boolean	**true** or false	When true, enforce the main audio buffer model When false, the main audio buffer may exceed buffer size depending on bitrate.
segmentation_time	float	> 0	The length in seconds of each segment. Required unless markers is set to *none*.
segmentation_style	string	maintain_cadence, reset_cadence	The segmentation style parameter controls how segmentation markers are inserted into the transport stream. With avails, it is possible that segments may be truncated, which can influence where future segmentation markers are inserted. When a segmentation style of “reset_cadence” is selected and a segment is truncated due to an avail, we will reset the segmentation cadence. This means the subsequent segment will have a duration of of $segmentation_time seconds. When a segmentation style of “maintain_cadence” is selected and a segment is truncated due to an avail, we will not reset the segmentation cadence. This means the subsequent segment will likely be truncated as well. However, all segments after that will have a duration of $segmentation_time seconds. Note that EBP lookahead is a slight exception to this rule.
fragment_time	float	>= 0	The length in seconds of each fragment. Only used with EBP markers.

NAME	TYPE	RANGE	DESCRIPTION
segmentation_markers	string	none, rai_segstart, rai_adapt, psi_segstart, ebp, ebp_legacy	Inserts segmentation markers at each segmentation_time period. rai_segstart sets the Random Access Indicator bit in the adaptation field. rai_adapt sets the RAI bit and adds the current timecode in the private data bytes. psi_segstart inserts PAT and PMT tables at the start of segments. ebp adds Encoder Boundary Point information to the adaptation field as per OpenCable specification OC-SP-EBP-I01-130118. ebp_legacy adds Encoder Boundary Point information to the adaptation field using a legacy proprietary format.
ebp_lookahead_ms	integer	0 – 10000 milliseconds	When set, enforces that Encoder Boundary Points do not come within the specified time interval of each other by looking ahead at input video. If another EBP is going to come in within the specified time interval, the current EBP is not emitted, and the segment is “stretched” to the next marker. The lookahead value does not add latency to the system. The AWS Elemental Live Event must be configured elsewhere to create sufficient latency to make the lookahead accurate.
ebp_on_audio	boolean		Controls placement of EBP on Audio PIDs. If checked, EBP markers will be placed on the video PID and all audio PIDs. If unchecked, EBP markers will be placed on only the video PID.
fixed_ebp_audio_interval	boolean		When this option is checked, audio EBP markers will be added to partitions 3 and 4. The interval between these additional markers will be fixed, and will be slightly shorter than the video EBP marker interval. Only available when EBP Cablelabs segmentation markers are selected.
force_ts_video_ebp_order	boolean		Select Force TS Video EBP Order to force Video EBP packets to appear before Audio EBP packets.
es_rate	boolean	true or **false**	Include the ES Rate field in the PES header.
arib	boolean	true or **false**	Enables ARIB-compliant field muxing and removes video descriptor.
dvb_nit_settings	DVB Network Information Table (NIT)	dvb_nit_settings	Inserts DVB Network Information Table (NIT) at the specified table repetition interval.
dvb_sdt_settings	DVB Service Description Table (SDT)	dvb_sdt_settings	Inserts DVB Service Description Table (NIT) at the specified table repetition interval.
dvb_tdt_settings	DVB Time and Date Table (TDT)	dvb_tdt_settings	Inserts DVB Time and Date Table (TDT) at the specified table repetition interval.
pmt_pid	integer	32 – 8182 (Default: **480**)	Packet Identifier (PID) for the Program Map Table (PMT) in the transport stream. Can be entered as a decimal or hexadecimal value.
pcr_pid	integer	32 – 8182	Packet Identifier (PID) of the Program Clock Reference (PCR) in the transport stream. When no value is given, the encoder will assign the same value as the Video PID. Can be entered as a decimal or hexadecimal value.
video_pid	integer	32 – 8182 (Default: **481**)	Packet Identifier (PID) of the elementary video stream in the transport stream. Can be entered as a decimal or hexadecimal value.
audio_pids	integer	32 – 8182 (Default: **482-498**)	Packet Identifier (PID) of the elementary audio stream(s) in the transport stream. Multiple values are accepted, and can be entered in ranges and/or by comma separation. Can be entered as decimal or hexadecimal values.
dvb_teletext_pid	integer	32 – 8182 (Default: **499**)	Packet Identifier (PID) for input source DVB Teletext data to this output. Can be entered as a decimal or hexadecimal value.
dvb_sub_pids	integer	32 – 8182 (Default: **460-479**)	Packet Identifier (PID) for input source DVB Subtitle data to this output. Multiple values are accepted, and can be entered in ranges and/or by comma separation. Can be entered as decimal or hexadecimal values.
scte35_pid	string	32 – 8182 (Default: **500**)	Packet Identifier (PID) of the SCTE-35 stream in the transport stream. Can be entered as a decimal or hexadecimal value.
scte35_esam_pid	integer	32 – 8182 (Default: **508**)	Packet Identifier (PID) of the SCTE-35 stream in the transport stream generated by the ESAM POIS. Can be entered as a decimal or hexadecimal value.
scte35_pullup	integer	0 – 8000 milliseconds (Default: **0**)	Pre-roll delay for SCTE-35 insertion in milliseconds. Only compatible with non-VBR SDI inputs. Zero means no delay.

NAME	TYPE	RANGE	DESCRIPTION
klv_data_pids	integer	32 – 8182 (Default: **501**)	Packet Identifier (PID) for input source KLV data to this output. Multiple values are accepted, and can be entered in ranges and/or by comma separation. Can be entered as decimal or hexadecimal values.
timed_metadata_pid	integer	32 – 8182 (Default: **502**)	Packet Identifier (PID) of the timed metadata stream in the transport stream. Can be entered as a decimal or hexadecimal value.
etv_platform_pid	integer	32 – 8182 (Default: **504**)	Packet Identifier (PID) for input source ETV Platform data to this output. Can be entered as a decimal or hexadecimal value.
etv_signal_pid	integer	32 – 8182 (Default: **505**)	Packet Identifier (PID) for input source ETV Signal data to this output. Can be entered as a decimal or hexadecimal value.
ecm_pid	integer	32 – 8182 (Default: **506**)	Packet Identifier (PID) for ECM in the transport stream. Only enabled when Simulcrypt is enabled. Can be entered as a decimal or hexadecimal value.

DVB NETWORK INFORMATION TABLE (NIT)

NAME	TYPE	RANGE	DESCRIPTION
rep_interval	integer	25 – 10000	The number of milliseconds between instances of this table in the output transport stream.
network_id	integer	0 – 65535	The numeric value placed in the Network Information Table (NIT).
network_name	string	1 – 256 characters	The network name text placed in the network_name_descriptor inside the Network Information Table. Maximum length is 256 characters.

DVB SERVICE DESCRIPTION TABLE (SDT)

NAME	TYPE	RANGE	DESCRIPTION
output_sdt	string	sdt_follow, sdt_follow_if_present, sdt_manual, sdt_none	Selects method of inserting SDT information into output stream. “Follow input SDT” copies SDT information from input stream to output stream. “Follow input SDT if present” copies SDT information from input stream to output stream if SDT information is present in the input, otherwise it will fall back on the user-defined values. Enter “SDT Manually” means user will enter the SDT information. “No SDT” means output stream will not contain SDT information.
rep_interval	integer	25 – 2000	The number of milliseconds between instances of this table in the output transport stream.
service_provider_name	string	1 – 256 characters	The service provider name placed in the service_descriptor in the Service Description Table. Maximum length is 256 characters.
service_name	string	1 – 256 characters	The service name placed in the service_descriptor in the Service Description Table. Maximum length is 256 characters.

DVB TIME AND DATE TABLE (SDT)

NAME	TYPE	RANGE	DESCRIPTION
rep_interval	integer	1000 – 30000	The number of milliseconds between instances of this table in the output transport stream.

SIMULCRYPT AES SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
channel_id	integer	>= 0	Integer value for the Verimatrix channel ID. Required.
ecmg	Location		Hostname or IP address for Veramatrix server, without tcp protocol details, and an optional port. Examples: tcp://12.34.56.78:1234, tcp://12.34.56.78, tcp://ecmg_host_name:1234, tcp://ecmg_host_name

NAME	TYPE	RANGE	DESCRIPTION
recommended_cp_duration	integer	>= 0	Desired minimum crypto-duration. May be overridden by the lower bound at the ECMG server.

M3U8 SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
pmt_pid	integer	32 – 8182 (Default: **480**)	Packet Identifier (PID) for the Program Map Table (PMT) in the transport stream. Can be entered as a decimal or hexadecimal value.
video_pid	integer	32 – 8182 (Default: **481**)	Packet Identifier (PID) of the elementary video stream in the transport stream. Can be entered as a decimal or hexadecimal value.
audio_pids	integer	32 – 8182 (Default: **482-498**)	Packet Identifier (PID) of the elementary audio stream(s) in the transport stream. Multiple values are accepted, and can be entered in ranges and/or by comma separation. Can be entered as decimal or hexadecimal values.
scte35_pid	string	32 – 8182 (Default: **500**)	Packet Identifier (PID) of the SCTE-35 stream in the transport stream. Can be entered as a decimal or hexadecimal value.
timed_metadata_pid	integer	32 – 8182 (Default: **502**)	Packet Identifier (PID) of the timed metadata stream in the transport stream. Can be entered as a decimal or hexadecimal value.
program_num	integer	0 – 65535 (Default **1**)	The value of the program number field in the Program Map Table.
pat_interval	integer	0, 10 – 1000 (Default: **0**)	The number of milliseconds between instances of this table in the output transport stream. A value of “0” writes out the PAT once per segment file.
pmt_interval	integer	0, 10 – 1000 (Default: **0**)	The number of milliseconds between instances of this table in the output transport stream. A value of “0” writes out the PMT once per segment file.
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 – 500	Maximum time in milliseconds between Program Clock References (PCRs) inserted into the transport stream.
pcr_pid	integer	32 – 8182	Packet Identifier (PID) of the Program Clock Reference (PCR) in the transport stream. When no value is given, the encoder will assign the same value as the Video PID. Can be entered as a decimal or hexadecimal value.
transport_stream_id	integer	0 – 65535	The value of the transport stream ID field in the Program Map Table.
audio_frames_per_pes	integer	>= 0 (Default: **4**)	The number of audio frames to insert for each PES packet.
ecm_pid	integer		ThePlatform-protected transport streams using ‘microsoft’ as Target Client include an ECM stream. This ECM stream contains the size, IV, and PTS of every sample in the transport stream. This stream PID is specified here. This PID has no effect on non ThePlatform-protected streams.
audio_buffer_compliant	boolean	**true** or false	When true, enforce the main audio buffer model. When false, the main audio buffer may exceed buffer size depending on bitrate.

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.

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	string	A valid Preset ID or name	Preset values to use for this stream. If this is included, no further parameters are needed. If video, audio, or caption parameters are included in the stream assembly, they will override the Preset value. A valid ID, name, or permalink should be provided.

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, h.265, mpeg2, frame capture, uncompressed, prores	Video codec. See Video Codecs for supported output codecs.
***codec*_settings**	Codec Settings	h264_settings, h265_settings, mpeg2_settings, frame_capture_settings, uncompressed_settings, prores_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.

NAME	TYPE	RANGE	DESCRIPTION
stretch_to_output	boolean	true or **false**	Automatically configures the output position Rectangle to stretch the video to the specified output resolution. This option will override any position value.
sharpness	integer	0: Softest – 100: Sharpest	Changes the width of the anti-alias filter kernel used for scaling. Only applies if scaling is being performed and anti_alias is set to true. 0 is the softest setting, 100 the sharpest, and 50 recommended for most content.
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**	A value of "true" passes through the selected timecode source value (in Timecode Config). This is only recommended when you are certain that input framerate is identical to output framerate. "False" removes the timecode from 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 possible 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.
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.
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	false or **true**	Includes colorspace metadata in the output.
insert_chroma_info	boolean	true or **false**	Enable this setting to include location information for chroma sampling. This setting applies to each output individually.
force_cpu_encode	boolean		Setting this control instructs the system to use a CPU encoder for this particular stream.

RECTANGLE

NAME	TYPE	RANGE	DESCRIPTION
x	integer		Left of rectangle. Specify only even numbers.
y	integer		Top of rectangle. Specify only even numbers.
width	integer		Width of rectangle in pixels. Specify only even numbers.
height	integer		Height of rectangle in pixels. Specify only even numbers.

H.264 SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
profile	enum	Baseline, **Main**, High, High 10-bit, High 4:2:2, High 4:2:2 10-bit	H.264 Profile. High 4:2:2 and 10-bit profiles are only available with the AVC-I License.

NAME	TYPE	RANGE	DESCRIPTION
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, QVBR	Rate control mode. CQ uses constant quantizer (qp), ABR (average bitrate) does not write HRD parameters. QVBR (quality-defined variable bitrate) should define quality level.
bitrate	integer	>= 1000 (Default: **5000000**)	Average bitrate in bits/second. Required for VBR, CBR, and ABR. Five megabits can be entered as 5000000 or 5m. Five hundred kilobits can be entered as 500000 or 0.5m. For MS Smooth outputs, bitrates must be unique when rounded down to the nearest multiple of 1000.
max_bitrate	integer		Maximum bitrate in bits/second. Applicable only to VBR and QVBR modes (required for QVBR). Five megabits can be entered as 5000000 or 5m. Five hundred kilobits can be entered as 500000 or 0.5m.
min_bitrate	integer		Optional. If specified, sets an explicit lower limit on the statmuxed instantaneous bitrate for this channel. If not specified, the minimum will be automatically set by the system.
buf_size	integer		Size of buffer (HRD buffer model). Five megabits can be entered as 5000000 or 5m. Five hundred kilobits can be entered as 500000 or 0.5m.
min_buf_occ	integer	>=0 to buf_size	Minimum occupancy of VBV / HRD buffer in bits.
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	This field applies only if the Streams > Advanced > Framerate (framerate) field is set to 29.970. This field works with the Streams > Advanced > Preprocessors > Deinterlacer field (deinterlace_mode) and the Streams > Advanced > Interlaced Mode field (interlace_mode) to identify the scan type for the output: Progressive, Interlaced, Hard Telecine or Soft Telecine. See Scan Types for information. - Hard: produces 29.97i output from 23.976 input. - Soft: produces 23.976; the player converts this output to 29.97i.
slow_pal	boolean	true or **false**	Enables Slow PAL rate conversion. 23.976fps and 24fps input is relabeled as 25fps, and audio is sped up correspondingly.
interlace_mode	enum	**progressive**, top_field, bottom_field, follow_top_field, follow_bottom_field	This field works with the Streams > Advanced > Preprocessors > Deinterlacer field (deinterlace_mode) and the Streams > Advanced > Telecine field (telecine) to identify the scan type for the output: Progressive, Interlaced, Hard Telecine or Soft Telecine. See Scan Types for information. The differences between the Top, Bottom, and Follow values are: - Top Field First or Bottom Field First produce interlaced with the entire output having the same field polarity (top or bottom first). - Follow (Default Top) and Follow (Default Bottom) use the same field polarity as the source. Therefore for the Follow options: - If the source is interlaced, the output will be interlaced with the same polarity as the source (it will follow the source). The output could therefore be a mix of "top field first" and "bottom field first". - If the source is progressive, the output will be interlaced with "top field first" or "bottom field first" polarity, depending on which of the Follow options you chose.
gop_size	float	> 0 (Default: **90**)	GOP Length (keyframe interval) in frames or seconds. Must be greater than zero.
gop_size_units	string	**frames** or seconds	Indicates if the GOP Size is specified in frames or seconds. If seconds the system will convert the GOP Size into a frame count at run time.
gop_num_b_frames	integer	0 – 7 (Default: **2**)	Number of B-frames between reference frames.

NAME	TYPE	RANGE	DESCRIPTION
dynamic_sub_gop	boolean	true or **false**	Enable this setting to improve subjective video quality for high-motion content. This will cause the product to use fewer B-frames (which infer information based on other frames) for high-motion portions of the video and more B-frames for low-motion portions. The maximum number of B-frames is limited by the value you provide for the setting B frames between reference frames (gop_num_b_frames).
repeat_pps	boolean	true or **false**	Places a PPS header on each encoded picture, even if repeated.
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**)	Enforces separation between repeated (cadence) I-frames and I-frames inserted by Scene Change Detection. If a scene change I-frame is within I-interval frames of a cadence I-frame, the GOP is shrunk and/or stretched to the scene change I-frame. GOP stretch requires enabling lookahead as well as setting I-interval. The normal cadence resumes for the next GOP. This setting is only used when Scene Change Detect is enabled. Note: Maximum GOP stretch = GOP size + Min-I-interval – 1
adaptive_quantization	string	off, low, medium, **high**, higher, max	Adaptive quantization. Allows intra-frame quantizers to vary to improve visual quality.
spatial_aq	boolean	**true** or false	Adjust quantization within each frame based on spatial variation of content complexity.
temporal_aq	boolean	**true** or false	Adjust quantization within each frame based on temporal variation of content complexity.
flicker_aq	boolean	true or **false**	Adjust quantization within each frame to reduce flicker or ‘pop’ on I-frames.
cabac	boolean	**true** or false	Enable CABAC (must be in Main or High profile).
softness	integer	0=default, 1=JVT, 16-128=planar interpolation	Softness. Selects quantizer matrix, larger values reduce high-frequency content in the encoded image.
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.
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	1,2,%(default)3%,4,5,6	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. When ‘Force Field Pictures’ is not enabled, the encoder may use either PAFF or MBAFF field/frame adaptation.
passes	integer	**1** or 2	Number of encoding passes.
gop_markers	boolean	true or **false**	Inserts a Recovery Point SEI message for open GOPs, or starts a new sequence for closed GOPs.
gop_b_reference	boolean	true or **false**	Enable use of reference B frames for GOP structures that have B frames > 1.
svq	integer	-2: Higher Quality, -1, **0**, 1, 2, 3: Higher Density	Selects encoding features based on performance. Higher values use fewer system resources so may allow more streams to be encoded.

NAME	TYPE	RANGE	DESCRIPTION
sei_timecode	boolean	true or **false**	Inserts timecode for each frame as 4 bytes of an unregistered SEI message.
rp2027_syntax	boolean		Produces a bitstream compliant with SMPTE RP-2027.

H.265 SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
profile	enum	**Main/Main**, Main/High, Main10/Main, Main10/High, Main 4:2:2 8-bit/Main, Main 4:2:2 8-bit/High, Main 4:2:2 10-bit/Main, Main 4:2:2 10-bit/High	Represents the Profile and Tier, per the HEVC (H.265) specification. Selections are grouped as [Profile] / [Tier], so “Main/High” represents Main Profile with High Tier. 4:2:2 profiles are only available with the HEVC 4:2:2 License.
level	enum	**Auto**, 1, 1.1, 2, 2.1, 3, 3.1, 4, 4.1, 5, 5.1, 5.2, 6, 6.1, 6.2	H.265 Level.
rate_control_mode	enum	VBR, **CBR**, CQ, ABR, QVBR	Rate control mode. CQ uses constant quantizer (qp), ABR (average bitrate) does not write HRD parameters. QVBR (quality-defined variable bitrate) should define quality level.
bitrate	integer	>= 1000 (Default: **5000000**)	Average bitrate in bits/second. Required for VBR, CBR, and ABR. Five megabits can be entered as 5000000 or 5m. Five hundred kilobits can be entered as 500000 or 0.5m. For MS Smooth outputs, bitrates must be unique when rounded down to the nearest multiple of 1000.
max_bitrate	integer		Maximum bitrate in bits/second. Applicable only to VBR and QVBR modes (required for QVBR). Five megabits can be entered as 5000000 or 5m. Five hundred kilobits can be entered as 500000 or 0.5m.
buf_size	integer		Size of buffer (HRD buffer model). Five megabits can be entered as 5000000 or 5m. Five hundred kilobits can be entered as 500000 or 0.5m.
buf_fill_pct	integer	0 – 100	Percentage of the buffer that should initially be filled (HRD buffer model).
min_i_interval	integer	0 – 30 (Default: **0**)	Enforces separation between repeated (cadence) I-frames and I-frames inserted by Scene Change Detection. If a scene change I-frame is within I-interval frames of a cadence I-frame, the GOP is shrunk and/or stretched to the scene change I-frame. GOP stretch requires enabling lookahead as well as setting I-interval. The normal cadence resumes for the next GOP. This setting is only used when Scene Change Detect is enabled. Note: Maximum GOP stretch = GOP size + Min-I-interval – 1
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.
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.
interpolate_frc	boolean	true or **false**	Interpolates during a framerate conversion. Produces smoother motion during a framerate change.
gop_size	float	> 0 (Default: **90**)	GOP Length (keyframe interval) in frames or seconds. Must be greater than zero.
gop_size_units	string	**frames** or seconds	Indicates if the GOP Size is specified in frames or seconds. If seconds the system will convert the GOP Size into a frame count at run time.
gop_num_b_frames	integer	0 – 7 (Default: **2**)	Number of B-frames between reference frames.
dynamic_sub_gop	boolean	true or **false**	Enable this setting to improve subjective video quality for high-motion content. This will cause the product to use fewer B-frames (which infer information based on other frames) for high-motion portions of the video and more B-frames for low-motion portions. The maximum number of B-frames is limited by the value you provide for the setting B frames between reference frames (gop_num_b_frames).

NAME	TYPE	RANGE	DESCRIPTION
slow_pal	boolean	true or **false**	Enables Slow PAL rate conversion. 23.976fps and 24fps input is relabeled as 25fps, and audio is sped up correspondingly.
repeat_pps	boolean	true or **false**	Places a PPS header on each encoded picture, even if repeated.
gop_b_reference	boolean	true or **false**	Enable use of reference B frames for GOP structures that have B frames > 1.
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.
qp	integer	1 – 51	Quantization parameter – fixed for CQ rate control mode, or starting QP for rate controller.
min_qp	integer	1 – 51	Minimum QP for rate controller.
max_qp	integer	1 – 51	Maximum QP 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.
tiles	boolean	**true** or false	Enable use of tiles, allowing horizontal as well as vertical subdivision of the encoded pictures.
adaptive_quantization	string	off, low, medium, **high**, higher, max	Adaptive quantization. Allows intra-frame quantizers to vary to improve visual quality.
spatial_aq	boolean	**true** or false	Adjust quantization within each frame based on spatial variation of content complexity.
temporal_aq	boolean	**true** or false	Adjust quantization within each frame based on temporal variation of content complexity.
flicker_aq	boolean	true or **false**	Adjust quantization within each frame to reduce flicker or ‘pop’ on I-frames.
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.
passes	integer	**1** or 2	Number of encoding passes.
svq	integer	-3: Higher Quality, -2, -1, **0**, 1, 2, 3: Higher Density	Selects encoding features based on performance. Higher values use fewer system resources so may allow more streams to be encoded. 0 is the lowest setting that will encode in real time for HD resolutions.
adaptive_sao	string	enabled, **adaptive**, off	Specify Sample Adaptive Offset (SAO) filter strength. Adaptive mode dynamically selects best strength based on content
alt_xfer_func_sei	boolean	true or **false**	Enables Alternate Transfer Function SEI message for outputs using Hybrid Log Gamma (HLG) Electro-Optical Transfer Function (EOTF).
interlace_mode	enum	**progressive**, top_field, bottom_field, follow_top_field, follow_bottom_field	This field works with the Streams > Advanced > Preprocessors > Deinterlacer field (deinterlace_mode) and the Streams > Advanced > Telecine field (telecine) to identify the scan type for the output: Progressive, Interlaced, Hard Telecine or Soft Telecine. See Scan Types for information. The differences between the Top, Bottom, and Follow values are: - Top Field First or Bottom Field First produce interlaced with the entire output having the same field polarity (top or bottom first). - Follow (Default Top) and Follow (Default Bottom) use the same field polarity as the source. Therefore for the Follow options: - If the source is interlaced, the output will be interlaced with the same polarity as the source (it will follow the source). The output could therefore be a mix of "top field first" and "bottom field first". - If the source is progressive, the output will be interlaced with "top field first" or "bottom field first" polarity, depending on which of the Follow options you chose.

NAME	TYPE	RANGE	DESCRIPTION
telecine	string	**None**, Soft, or Hard	This field applies only if the Streams > Advanced > Framerate (framerate) field is set to 29.970. This field works with the Streams > Advanced > Preprocessors > Deinterlacer field (deinterlace_mode) and the Streams > Advanced > Interlaced Mode field (interlace_mode) to identify the scan type for the output: Progressive, Interlaced, Hard Telecine or Soft Telecine. See Scan Types for information. - Hard: produces 29.97i output from 23.976 input. - Soft: produces 23.976; the player converts this output to 29.97i.
temporal_ids	boolean	true or **false**	Enables temporal layer identifiers in the encoded bitstream. Up to 3 layers are supported depending on GOP structure: I- and P-frames form one layer, reference B-frames can form a second layer and non-reference b-frames can form a third layer. Decoders can optionally decode only the lower temporal layers to generate a lower frame rate output. For example, given a bitstream with temporal IDs and with b-frames = 1 (i.e. IbPbPb display order), a decoder could decode all the frames for full frame rate output or only the I and P frames (lowest temporal layer) for a half frame rate output.
sei_timecode	boolean	true or **false**	Inserts timecode for each frame as 4 bytes of an unregistered SEI message.

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. QVBR (quality-defined variable bitrate) is not allowed in mpeg2 codec.
bitrate	integer	>= 1000 (Default: **5000000**)	Average bitrate in bits/second. Required for VBR, CBR, and ABR. Five megabits can be entered as 5000000 or 5m. Five hundred kilobits can be entered as 500000 or 0.5m. For MS Smooth outputs, bitrates must be unique when rounded down to the nearest multiple of 1000.
intra_dc_precision	enum	**auto**, 8, 9, 10, 11	Select quantization precision for intra-block DC coefficients. Auto selects precision based on per-frame compression ratio, other selections set precision to a fixed value.
max_bitrate	integer		Maximum bitrate in bits/second (for VBR mode only). Five megabits can be entered as 5000000 or 5m. Five hundred kilobits can be entered as 500000 or 0.5m.
buf_size	integer		Size of buffer (HRD buffer model). Five megabits can be entered as 5000000 or 5m. Five hundred kilobits can be entered as 500000 or 0.5m.
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	This field applies only if the Streams > Advanced > Framerate (framerate) field is set to 29.970. This field works with the Streams > Advanced > Preprocessors > Deinterlacer field (deinterlace_mode) and the Streams > Advanced > Interlaced Mode field (interlace_mode) to identify the scan type for the output: Progressive, Interlaced, Hard Telecine or Soft Telecine. See Scan Types for information. - Hard: produces 29.97i output from 23.976 input. - Soft: produces 23.976; the player converts this output to 29.97i.

NAME	TYPE	RANGE	DESCRIPTION
slow_pal	boolean	true or **false**	Enables Slow PAL rate conversion. 23.976fps and 24fps input is relabeled as 25fps, and audio is sped up correspondingly.
interlace_mode	enum	**progressive**, top_field, bottom_field, follow_top_field, follow_bottom_field	This field works with the Streams > Advanced > Preprocessors > Deinterlacer field (deinterlace_mode) and the Streams > Advanced > Telecine field (telecine) to identify the scan type for the output: Progressive, Interlaced, Hard Telecine or Soft Telecine. See Scan Types for information. The differences between the Top, Bottom, and Follow values are: - Top Field First or Bottom Field First produce interlaced with the entire output having the same field polarity (top or bottom first). - Follow (Default Top) and Follow (Default Bottom) use the same field polarity as the source. Therefore for the Follow options: - If the source is interlaced, the output will be interlaced with the same polarity as the source (it will follow the source). The output could therefore be a mix of "top field first" and "bottom field first". - If the source is progressive, the output will be interlaced with "top field first" or "bottom field first" polarity, depending on which of the Follow options you chose.
gop_size	float	> 0 (Default: **12**)	GOP Length (keyframe interval) in frames or seconds. Must be greater than zero.
gop_size_units	string	**frames** or seconds	Indicates if the GOP Size is specified in frames or seconds. If seconds the system will convert the GOP Size into a frame count at run time.
gop_num_b_frames	integer	0 – 7 (Default: **2**)	Number of B-frames between reference frames.
dynamic_sub_gop	boolean	true or **false**	Enable this setting to improve subjective video quality for high-motion content. This will cause the product to use fewer B-frames (which infer information based on other frames) for high-motion portions of the video and more B-frames for low-motion portions. The maximum number of B-frames is limited by the value you provide for the setting B frames between reference frames (gop_num_b_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**)	Enforces separation between repeated (cadence) I-frames and I-frames inserted by Scene Change Detection. If a scene change I-frame is within I-interval frames of a cadence I-frame, the GOP is shrunk and/or stretched to the scene change I-frame. GOP stretch requires enabling lookahead as well as setting I-interval. The normal cadence resumes for the next GOP. This setting is only used when Scene Change Detect is enabled. Note: Maximum GOP stretch = GOP size + Min-I-interval – 1
adaptive_quantization	string	off, low, medium, **high**, higher, max	Adaptive quantization. Allows intra-frame quantizers to vary to improve visual quality.
spatial_aq	boolean	**true** or false	Adjust quantization within each frame based on spatial variation of content complexity.
temporal_aq	boolean	**true** or false	Adjust quantization within each frame based on temporal variation of content complexity.
framing_aq	boolean	true or **false**	Decrease inter-frame quantization in the center of the frame, increase near the edges.
softness	integer	0=default, 16-128=planar interpolation	Softness. Selects quantizer matrix, larger values reduce high-frequency content in the encoded image.
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.
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.

NAME	TYPE	RANGE	DESCRIPTION
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.
min_buf_occ	integer	>=0 to buf_size	Minimum occupancy of VBV / HRD buffer in bits.

PRORES 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.
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.
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	This field works with the Streams > Advanced > Preprocessors > Deinterlacer field (deinterlace_mode) and the Streams > Advanced > Telecine field (telecine) to identify the scan type for the output: Progressive, Interlaced, Hard Telecine or Soft Telecine. See Scan Types for information. The differences between the Top, Bottom, and Follow values are: - Top Field First or Bottom Field First produce interlaced with the entire output having the same field polarity (top or bottom first). - Follow (Default Top) and Follow (Default Bottom) use the same field polarity as the source. Therefore for the Follow options: - If the source is interlaced, the output will be interlaced with the same polarity as the source (it will follow the source). The output could therefore be a mix of "top field first" and "bottom field first". - If the source is progressive, the output will be interlaced with "top field first" or "bottom field first" polarity, depending on which of the Follow options you chose.
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	This field applies only if the Streams > Advanced > Framerate (framerate) field is set to 29.970. This field works with the Streams > Advanced > Preprocessors > Deinterlacer field (deinterlace_mode) and the Streams > Advanced > Interlaced Mode field (interlace_mode) to identify the scan type for the output: Progressive, Interlaced, Hard Telecine or Soft Telecine. See Scan Types for information. - Hard: produces 29.97i output from 23.976 input. - Soft: produces 23.976; the player converts this output to 29.97i.
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_dropdown	enum	Every 5 seconds, Middle frame, At 20 seconds or fallback to middle, Custom.	Choose from a range of predefined common frame capture timing instructions.
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.
fourcc	string		Choose the FourCC code that is appropriate for the intended downstream client player. Note that some client players may not play video if the FourCC code used is not one of the supported codes of that player.

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, force_hdr10, force_hlg_2020	Determines if colorspace conversion will be performed. If set to *None*, no conversion will be performed. If any of the *Force* options are selected, the input will be converted to the specified color space. 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, PNG or TGA 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*.

NAME	TYPE	RANGE	DESCRIPTION
enable_rest	boolean	true or **false**	Indicates that REST commands will be used to send image insertion commands. If used, no other fields are needed.
insertable_image	Insertable Image		Image to insert. Must be 32 bit windows BMP, PNG, or TGA file. Must not be larger than the output frames.

INSERTABLE IMAGE

NAME	TYPE	RANGE	DESCRIPTION
image_inserter_input	Location		Image to insert. Must be 32 bit windows BMP, PNG or TGA. Must not be larger than the output frames.
layer	integer	0 – 99	The Z order of the inserted image. Images with higher values of layer will be inserted on top of images with lower values of layer.
image_x	integer		Placement of image on the horizontal axis in pixels. 0 is the left edge of the frame. Required for BMP, PNG and TGA input.
image_y	integer		Placement of image on the vertical axis in pixels. 0 is the top edge of the frame. Required for BMP, PNG and TGA input.
opacity	integer	0 – 100 (Default: **50**)	Opacity of image. 0 is transparent. 100 is fully opaque. Required for BMP, PNG and TGA input.
width	integer		The width of the image when inserted in the video. Leave blank to use the native width of the image.
height	integer		The height of the image when inserted in the video. Leave blank to use the native height of the image.
start_time	string		The start time for the image. Must be in timecode (HH:MM:SS:FF) format.
duration	integer		The time in milliseconds for the image to remain in the video.
fade_in	integer		The time in milliseconds for the image to fade in.
fade_out	integer		The time in milliseconds for the image to fade out.

DEINTERLACER

NAME	TYPE	RANGE	DESCRIPTION
deinterlace_mode	string	**Deinterlace**, Inverse Telecine, Adaptive	This field works with the Streams > Advanced > Telecine field (telecine) and the Streams > Advanced > Interlaced Mode field (interlace_mode) to identify the scan type for the output: Progressive, Interlaced, Hard Telecine or Soft Telecine. See Scan Types for information. - Deinterlace converts interlaced to progressive. - Inverse Telecine converts Hard Telecine 29.97i to progressive 23.976p. - Adaptive auto-detects and converts to progressive.

NAME	TYPE	RANGE	DESCRIPTION
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.
force	boolean	true or **false**	This field appears only when Streams > Advanced > Preprocessors > Deinterlacer field (deinterlace_mode) is enabled: - When Force Mode is off (default), the processor does not convert frames that are tagged in metadata as progressive. It will only convert those that are tagged as some other type. - When Force Mode is on, the processor converts every frame to progressive, even those that are already tagged as progressive. Turn Force mode on only if there is a good chance that the metadata has tagged frames as progressive when they are not progressive. Do not turn on otherwise; processing frames that are already progressive into progressive will probably result in lower quality video.

NOISE REDUCER

NAME	TYPE	RANGE	DESCRIPTION
filter	string	**Bilateral**, Mean, Gaussian, Lanczos, Sharpen, Conserve, Spatial	Filters perform a range of spatial image filtering functions Bilateral is an edge preserving noise reduction filter; Mean (softest), Gaussian, Lanczos, and Sharpen (sharpest) are convolution filters; Conserve is a min/max noise reduction filter
strength	integer	0-16 for Spatial filter, 0-3 for other filters	Relative strength of filtering (higher produces stronger filtering). The range for the Spatial filter is 0-16, 4 is the default. The range for the other filters is 0-3, 1 is the default.
speed	integer	-2 to 3	The speed of the filter, from -2 (lower speed) to 3 (higher speed), with 0 being the nominal value. Only applicable to Spatial filter.
post_filter_sharpen_strength	integer	0 to 3	Specify strength of post noise reduction sharpening filter, with 0 disabling the filter and 3 enabling it at maximum strength. Only applicable to Spatial filter.

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, Pay TV C	Specifies a 3rd party watermarking provider. Currently, only Civolution and Pay-TV-C are 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.

NAME	TYPE	RANGE	DESCRIPTION
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, 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, pass_through_settings, mp2_settings, ac3_settings, eac3_settings, dtse_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. The ISO 639 language specified in the 'Language Code' drop down will be used when 'Follow Input Language Code' is not selected or when 'Follow Input Language Code' is selected but there is no ISO 639 language code specified by the input.
stream_name	string	Alphanumeric characters, spaces, and underscore	Used for MS Smooth and Apple HLS 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.
audio_normalization_settings	Audio Normalization Settings		Advanced audio normalization settings.

AAC SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
bitrate	integer		Average bitrate in bits/second. Valid values depend on rate control mode and profile.
sample_rate	integer		Sample rate in hz. Valid values depend on rate control mode and profile.
coding_mode	string	1_0, 1_1, ad_receiver_mix, **2_0**, 5_1	Mono (Audio Description), Mono, Stereo, or 5.1 channel layout. Valid values depend on rate control mode and profile. "1.0 – Audio Description (Receiver Mix)" setting receives a stereo description plus control track and emits a mono AAC encode of the description track, with control data emitted in the PES header as per ETSI TS 101 154 Annex E.
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 containers.

NAME	TYPE	RANGE	DESCRIPTION
mpeg2	boolean	true or **false**	Use MPEG-2 AAC audio instead of MPEG-4 AAC audio for raw or MPEG-2 Transport Stream containers.
ad_broadcaster_mix	boolean	true or **false**	Check (set to true) when input contains pre-mixed main audio + AD (narration) as a stereo pair. The Audio Type field (audio_type) will be set to 3, which signals to downstream systems that this stream contains “broadcaster mixed AD”. Note that the input received by the encoder must contain pre-mixed audio; the encoder does not perform the mixing. The values in Follow Audio Input Type and Audio Type are ignored. Leave unchecked (set to false) when input does not contain pre-mixed audio + AD. In this case, complete Follow Audio Input Type and Audio Type as desired.
align	boolean	true or **false**	Enable this setting to force the AAC preroll to a fixed length. This ensures AAC frame alignment across AAC outputs of the same AAC profile.
vbr_quality	enum	LOW, MEDIUM_LOW, MEDIUM_HIGH, HIGH	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**, 4, 8	Mono, Stereo, 4-Channel, or 8-Channel.
bit_depth	integer	**16** or 24	Bits per sample.
rf64	boolean		Indicate the format in WAVE header. If the raw audio stream is larger than 4 GiB, it needs to be put in RF64 format. Checked: RF64 format Unchecked: RIFF format

AIFF SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
sample_rate	integer	8000 – 192000 (Default: **44100**)	Sample rate in hz.
channels	integer	1, **2**, 6	Mono, Stereo, or 5.1.
bit_depth	integer	**16** or 24	Bits per sample.

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	64000 – 640000 (Default: **192000**)	Average bitrate in bits/second. Valid bitrates depend on the coding mode.

NAME	TYPE	RANGE	DESCRIPTION
sample_rate	integer	Sample rate is always **48000**	Sample rate in hz.
bitstream_mode	string	**complete_main**, commentary, dialogue, emergency, hearing_impaired, music_and_effects, visually_impaired, voice_over	Specifies the “Bitstream Mode” (bsmod) for the emitted AC-3 stream. See ATSC A/52-2012 for background on these values.
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.
follow_input_metadata	boolean	true or **false**	When true, Encoder metadata will be sourced from the DD, DD+, or DolbyE decoder that supplied this audio data. If audio was not supplied from one of these streams, then the above static metadata settings will be used.

DOLBY DIGITAL PLUS AUDIO SETTINGS

Requires license

NAME	TYPE	RANGE	DESCRIPTION
bitrate	integer	64000 – 640000 (Default: **192000**)	Average bitrate in bits/second. Valid bitrates depend on the coding mode.
sample_rate	integer	Sample rate is always **48000**	Sample rate in hz.
bitstream_mode	string	**complete_main**, commentary, emergency, hearing_impaired, visually_impaired	Specifies the “Bitstream Mode” (bsmod) for the emitted E-AC-3 stream. See ATSC A/52-2012 (Annex E) for background on these values.
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.
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.

NAME	TYPE	RANGE	DESCRIPTION
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.
follow_input_metadata	boolean	true or **false**	When true, Encoder metadata will be sourced from the DD, DD+, or DolbyE decoder that supplied this audio data. If audio was not supplied from one of these streams, then the above static metadata settings will be used.
passthrough_when_possible	boolean	true or **false**	When checked, input DD+ audio will be passed through if it is present on the input. This detection is dynamic over the life of the transcode. Inputs that alternate between DD+ and non-DD+ content will have a consistent DD+ output as the system alternates between passthrough and encoding.

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**, 4, 6, 8, or 1-16	Number of output channels to be produced. Ranges will vary based on whether the remix is input-based (1-16) or output-based (1,2,4,6,8).
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.
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.
correction_gate_level	integer	-70 to 0	Content measuring above this level will be corrected to the target level. Content measuring below this level will not be corrected. Gating only applies when not using real_time_correction.

CAPTION DESCRIPTION

NAME	TYPE	RANGE	DESCRIPTION
order	integer	> 0	Required for multiple captions. Specifies the order the caption descriptions should be listed in.
caption_source_name	string		Specifies which Caption Selector to use from each input when generating captions. The name should be of the format "Caption Selector <N>", which denotes that the Nth Caption Selector will be used from each input.
destination_type	string	ARIB, Burn-In, CFF-TT, DVB-Sub, **Embedded**, Embedded+SCTE-20, SCTE-20+Embedded, SCC, SMI, SMPTE-TT, SRT, Teletext, TTML, WebVTT	Destination format for captions. Captions with an external file destination must be specified using a separate caption-only output. Embedded captions in a Quicktime container result in a caption track.
***destination*_settings**	Destination Settings	burnin_destination_settings, scc_destination_settings, dvb_sub_destination_settings, teletext_destination_settings, embedded_destination_settings	Specific settings required by destination type. Note that burnin_destination_settings are not available if the source of the caption data is Embedded or Teletext.
style_passthrough	boolean	**true** or false	Pass through style and position information from a TTML-like input source (TTML, SMPTE-TT, CFF-TT) to the CFF-TT output or TTML output.
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	Human readable information to indicate captions available for players (eg. English, or Spanish).

MS SMOOTH TTML TEMPLATES

There are two template files used by AWS Elemental Conductor File 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.

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’. Although the user can select output fonts for many different types of input captions, embedded, STL and teletext sources use a strict grid system. Using external fonts with these caption sources could cause unexpected display of proportional fonts. All burn-in and DVB-Sub font settings must match. Any italics, bold, or bold-italics versions of these fonts must be placed in the same directory as the main external font. They must also be named as follows: normal => fontname.ttf italics => fontname-Oblique.ttf or fontnameOblique.ttf bold => fontname-Bold.ttf or fontnameBold.ttf bold + italics => fontname-BoldOblique.ttf or fontnameBoldOblique.ttf
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. All burn-in and DVB-Sub font settings must match.
font_resolution	integer	**96** – 600	Font resolution in DPI (dots per inch); default is 96 dpi. All burn-in and DVB-Sub font settings must match.
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. All burn-in and DVB-Sub font settings must match.
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. All burn-in and DVB-Sub font settings must match.
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. All burn-in and DVB-Sub font settings must match.
teletext_fixed_grid	boolean	**true**, false	Controls whether a fixed grid size will be used to generate the output subtitles bitmap. Only applicable for Teletext or STL inputs and DVB-Sub/ Burn-in outputs.

NAME	TYPE	RANGE	DESCRIPTION
font_color	string	**white**, black, yellow, red, green, blue, black	Specifies the color of the burned-in captions. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.
font_opacity	integer	0 – **255**	Specifies the opacity of the burned-in captions. 255 is opaque; 0 is transparent. All burn-in and DVB-Sub font settings must match.
background_color	string	**none**, black, white	Specifies the color of the rectangle behind the captions. All burn-in and DVB-Sub font settings must match.
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). All burn-in and DVB-Sub font settings must match.
outline_size	integer	0 – 10	Specifies font outline size in pixels. This option is not valid for source captions that are either 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.
outline_color	string	**black**, white, yellow, red, green, blue	Specifies font outline color. This option is not valid for source captions that are either 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.
shadow_color	string	**none**, black, white	Specifies the color of the shadow cast by the captions. All burn-in and DVB-Sub font settings must match.
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). All burn-in and DVB-Sub font settings must match.
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. All burn-in and DVB-Sub font settings must match.
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 above the text. All burn-in and DVB-Sub font settings must match.

SCC DESTINATION SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
framerate	string	23.97, 24, 29.97 dropframe, 29.97 non-dropframe	Complete this field to ensure that the captions and the video are synchronized in the output. If video framerate is 24 or 23.976, choose 29.97 non-dropframe unless the workflow demands otherwise. If the video framerate is 29.97, choose 29.97 dropframe only if the video has video_insertion=true and drop_frame_timecode true; otherwise, choose 29.97 non-dropframe.

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.

NAME	TYPE	RANGE	DESCRIPTION
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, raw, m2ts, m3u8, ismv, 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	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.
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.
insert_scte35_esam	boolean	true or **false**	If true, update any SCTE-35 signals from ESAM POIS to this output. Only available for m2ts 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.
insert_timed_metadata	boolean	true or **false**	If true, inserts ID3 timed metadata from the timed_metadata REST command into this 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.

NAME	TYPE	RANGE	DESCRIPTION
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.
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.
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.
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.
avsync_enable	boolean	**true** or false	Enables A/V sync.
avsync_pad_trim_audio	boolean	**true** or false	Enables A/V sync trim audio.
nielsen_configuration	Nielsen Configuration		Nielsen configuration settings
output_group	Output Group		An output group for this Profile. Output groups contain information about where streams should be distributed.
ad_avail_offset	integer	-1000 – 1000 (Default: **0**)	When specified, this offset (in milliseconds) is added to the input Ad Avail PTS time.

JOB WATCH FOLDER

NAME	TYPE	RANGE	DESCRIPTION
incoming	Location		Folder to watch for new content. URI must be a path or valid S3 URI. Remote watch folders may require credentials.
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 (20 – 300 for remote) (Default: **2**)	Polling interval in seconds to detect whether the file size continues to grow. It is recommended to increase this interval if Jobs begin before files copy. Note: remote watch folders have a min of 20s
active	boolean	**true** or false	Whether this Watch Folder is currently active.
search_subfolders	boolean	true or **false**	Search subfolders for content

SEQUENCER CONFIG

NAME	TYPE	RANGE	DESCRIPTION
pct_rt_threshold	integer		The maximum percent of resource utilization to still be considered realtime processing. Lower percentages allow more jobs to run simultaneously, but consume more resources and can delay individual job runtime.

NAME	TYPE	RANGE	DESCRIPTION
use_cpu_size	integer		Offloads a small task to the CPU if its output resolution (width * height) is less than or equal to this value and use_cpu_rate conditions are met
use_cpu_rate	integer		Offloads a small task to the CPU if its target bitrate is less than or equal to this value and use_cpu_size conditions are met
cpu_load_factor	integer		The maximum number of threads to utilize per CPU. This affects the maximum amount of work the system will execute on the CPU simultaneously.
use_cpu_saturated	boolean		Allows offloading additional tasks to the CPU if the GPU is saturated
job_poll_rate	integer		How often (in seconds) to check the status of existing jobs
watchfolder_poll_rate	integer		How often (in seconds) to check the watch folder for new files
stalejob_poll_rate	integer		How often (in seconds) to check if any jobs are stale
jobcancelled_poll_rate	integer		How often (in seconds) to check if any jobs have been canceled
max_jobs	integer		The maximum number of jobs that can be running simultaneously
exclude_gpu_0	boolean		Prevents jobs from running on GPU 0
exclude_gpu_1	boolean		Prevents jobs from running on GPU 1
exclude_gpu_2	boolean		Prevents jobs from running on GPU 2
exclude_gpu_3	boolean		Prevents jobs from running on GPU 3
stalejob_age	integer		How long (in seconds) since the last update to consider a job to be stale
media_info_timeout	integer		Number of seconds before elemental_se quits trying to determine an input’s media info during preprocessing
disable_profiles_and_levels_messaging	boolean		Show [Profiles and Levels] type messaging during processing. Retrieving messages via REST is not affected by this setting.

CLUSTER SEQUENCER CONFIG

NAME	TYPE	RANGE	DESCRIPTION
pct_rt_threshold	integer		The maximum percent of resource utilization to still be considered realtime processing. Lower percentages allow more jobs to run simultaneously, but consume more resources and can delay individual job runtime.
use_cpu_size	integer		Offloads a small task to the CPU if its output resolution (width * height) is less than or equal to this value and use_cpu_rate conditions are met
use_cpu_rate	integer		Offloads a small task to the CPU if its target bitrate is less than or equal to this value and use_cpu_size conditions are met
cpu_load_factor	integer		The maximum number of threads to utilize per CPU. This affects the maximum amount of work the system will execute on the CPU simultaneously.
use_cpu_saturated	boolean		Allows offloading additional tasks to the CPU if the GPU is saturated
job_poll_rate	integer		How often (in seconds) to check the status of existing jobs
watchfolder_poll_rate	integer		How often (in seconds) to check the watch folder for new files
stalejob_poll_rate	integer		How often (in seconds) to check if any jobs are stale
jobcancelled_poll_rate	integer		How often (in seconds) to check if any jobs have been canceled
max_jobs	integer		The maximum number of jobs that can be running simultaneously
copy_local_dir	string		When the copy_local flag is set for a job, all of its file inputs will be copied into this working directory before the transcode begins. Non-streaming network file inputs (see URI types) are always downloaded here regardless of the copy_local flag.
exclude_gpu_0	boolean		Prevents jobs from running on GPU 0
exclude_gpu_1	boolean		Prevents jobs from running on GPU 1
exclude_gpu_2	boolean		Prevents jobs from running on GPU 2

NAME	TYPE	RANGE	DESCRIPTION
exclude_gpu_3	boolean		Prevents jobs from running on GPU 3
stalejob_age	integer		How long (in seconds) since the last update to consider a job to be stale
media_info_timeout	integer		Number of seconds before elemental_se quits trying to determine an input's media info during preprocessing

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, except when Rate Control Mode is set to QVBR, in which case this field resolves to the value in Max 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).
ex	Extension	Extension of the output file
job	Job ID	ID of the AWS Elemental Server job. Used to ensure unique output destinations.
$event$	Event ID	ID of the AWS Elemental Live event. Used to ensure unique output destinations.
$$	$	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

Name Modifiers for DASH ISO outputs accept two format identifiers: $Number$ and $Bandwidth$. In the manifest, $Number$ allows SegmentTemplate to contain "duration" and "startNumber".

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
```

SCAN TYPES

This table describes how to convert progressive, interlace, or telecine input to a different scan type in the output.

INPUT AND DESIRED OUTPUT		HOW TO GET THERE			
INPUT	OUTPUT	PREPROCESSOR FIELD	FORCE MODE FIELD (FORCE)	INTERLACE MODE FIELD	TELECINE MODE FIELD
Progressive	Progressive	Off	n/a	Progressive	None
Interlaced	Progressive	Deinterlace	On: if you know that metadata is tagged as progressive when in fact it is not progressive. Off: if frames are correctly tagged.	Progressive	None
Interlaced	Progressive	Adaptive	Off	Progressive	None
Hard telecine	Progressive	Inverse telecine	On: if you know that metadata is tagged as progressive when in fact it is not progressive. Off: if frames are correctly tagged.	Progressive	None
Hard telecine	Progressive	Adaptive	Off	Progressive	None
Soft telecine	Progressive	Off	n/a	Progressive	None
Mixed	Progressive	Adaptive	Off	Progressive	None
Progressive	Hard telecine	Off	n/a	Interlace	Hard telecine
Hard telecine	Hard telecine	Off	n/a	Interlace	None
Soft telecine	Hard telecine	Off	n/a	Interlace	Hard telecine
Mixed	Hard telecine	Off	n/a	Interlace	Hard telecine
Interlaced	Interlaced	Off	n/a	Interlace	None
Mixed	Interlaced	Off	n/a	Interlace	None
Progressive	Soft telecine	Off	n/a	Interlace	Soft telecine
Hard telecine	Soft telecine	Inverse telecine	On: if you know that metadata is tagged as progressive when in fact it is not progressive. Off: if frames are correctly tagged.	Interlace	Soft telecine
Hard telecine	Soft telecine	Adaptive	Off	Interlace	Soft telecine
Soft telecine	Soft telecine	Off	n/a	Interlace	Soft telecine
Mixed	Soft telecine	Adaptive	Off	Interlace	Soft telecine

SNMP INTERFACE

The AWS Elemental Conductor File 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. Individual nodes in the cluster can also be monitored independently through SNMP. The operator should enable SNMP access on each node in its settings page.

A user can interact with the system using a variety of network management systems. AWS Elemental Conductor File 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
- Operations
  - Base SNMP Operations
  - Job Operations
- SNMP Traps

SNMP BASICS

External access to the SNMP interface can be enabled in the Cluster -> 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 AWS 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 AWS 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 AWS 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
```

AWS 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 AWS Elemental products
- http://<server_ip>/mib/ELEMENTAL_SERVER_MIB.txt - Objects specific to AWS Elemental Server nodes

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 AWS Elemental Conductor File 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 database, this procedure could take some time and frequent polling can affect system performance.

SNMP OPERATIONS

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

VARIABLE	TYPE	GET VALUES	SET VALUES
ELEMENTAL-MIB::serviceStatus	Integer	0 if the elemental_se service is not running, 1 if the service is running	0 stops the elemental_se service. 1 starts the service, and 2 restarts the service
ELEMENTAL-MIB::firewallStatus	Integer	0 if the system's firewall is off, 1 if on	1 will load new firewall settings. Firewall settings are configured in the AWS 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 AWS 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 AWS Elemental web interface.
ELEMENTAL-MIB::version	String	Product version	
ELEMENTAL-MIB::httpdStatus	Integer	0 if the httpd service is not running, 1 if the service is running	0 stops the httpd service. 1 starts the service, and 2 restarts the service
ELEMENTAL-MIB::databaseBackup	Integer	1 if writes (starting backups) are allowed. 0 if writes are not allowed	1 starts a database backup, any other value in a SET command is an error.

SNMP JOB OPERATIONS

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
ELEMENTAL-SERVER-MIB::nodeId	Integer	The numerical ID of the node the job is running on

SNMP TRAPS

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

SNMP Traps are generated for the following events:

NOTIFICATION	EVENT	CONTENTS
ELEMENTAL-MIB::alert	Any alert generated by the system	ELEMENTAL-MIB::alertSet: 1 if the alert is being set, 0 if the alert is being cleared ELEMENTAL-MIB::alertMessage: Message describing the alert that was set or cleared ELEMENTAL-MIB::alertNodeId: The numerical ID of the node generating the alert. ELEMENTAL-MIB::alertRunnableId: The numerical ID of the Job or AWS Elemental Live Event generating the alert, if applicable.

AUTHENTICATION

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

The AWS Elemental Conductor File 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 AWS Elemental Conductor File 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 AWS Elemental Conductor File 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.

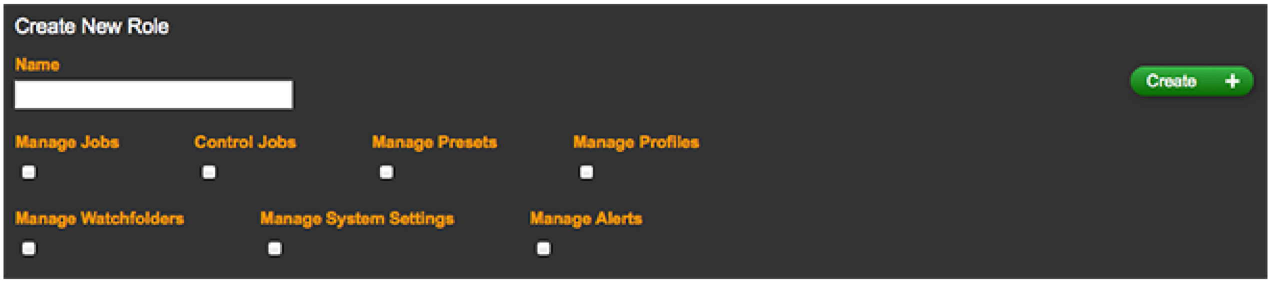
The AWS Elemental Conductor File system comes with a set of predefined Roles:

- **Admin**: The Admin role has access to the entire AWS Elemental Conductor File 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 AWS Elemental Conductor File 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 AWS 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 Password Expires field allows a user to be created with a password that will automatically expire after a set period of time. The Force Password Reset checkbox will force the user to reset their password the first time they login.

Role: Manager

Expires: Never

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

USER PROFILE

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

AUTHENTICATION AND REST

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

REFERENCE

- Supported Codecs and Containers
- Supported HLS Player Versions
- Supported Caption Formats

SUPPORTED CODECS AND CONTAINERS

NOTES

CODEC OR CONTAINER	DIRECTION	STATEMENT
MXF input container for video	Input	Complete list of supported containers is: AS-02, OP-1a; OP-1b; OP-1c; OP-2a ; OP-2b ; OP-2c; OP-3c.
IMF composition playlist (CPL)	Input	Complete list of supported IMF specializations is App #2, and App #2e
Apple® ProRes video codec	Input	Complete list of supported codecs is: Apple Prores 444 (all profiles); Apple Prores 4444 (all profiles); Apple Prores 422 (all profiles). Apple Prores 444 and 4444 will be converted to Apple Prores 422 during input handling.
MPEG-2 video codec	Input	Complete list of supported codecs is: MPEG-2; ATSC (A/53).
AAC audio codec	Input	Complete list of supported profiles is: LC-AAC, HE-AAC v1 and HE-AAC v2.
Dolby® Digital audio codec	Input	Dolby Digital is also known as AC-3 Dolby Digital is a licensed codec; however, no license is required to decode this codec in input.
Dolby® Digital Plus™ audio codec	Input	Dolby Digital Plus™ is also known as Enhanced AC-3 and is frequently abbreviated as DD+ or EC-3 and E-AC-3 Decoding of Dolby Digital Plus™ requires the AWS Elemental Audio Decode Package license option.
Dolby® E frames carried in PCM audio streams	Input	Decoding of Dolby E in PCM stream requires the AWS Elemental Audio Decode Package license option
MPEG Audio codec	Input	Complete list of supported codecs is: MPEG-1 Audio Layer II; MPEG-2 Audio Layer II (also known as MP2); MPEG-1 Audio Layer III (also known as MP3).
Apple® ProRes video codec in output	Output	Complete list of supported codecs is: Apple Prores 422 (all profiles).
Dolby® Digital audio codec	Output	Encoding with Dolby Digital requires the AWS Elemental Advanced Audio Package license option.
Dolby® Digital Plus audio codec	Output	Encoding with Dolby Digital Plus™ requires the AWS Elemental Advanced Audio Package license option.
Dolby® E pass-through	Output	See the last page of this document.
DTS Express™	Output	Encoding with DTS Express requires the AWS Elemental Advanced Audio Package license option.

CONTAINERS AND CODECS FOR FILE INPUTS

CONTAINER	MEDIA TYPE	EXTENSIONS	VIDEO CODECS	AUDIO CODECS
No Container		.m2v, .m1v	DV/DVCPRO H.264 HEVC (H.265) MPEG-1 MPEG-2	
Apple® HTTP Live Streaming	HLS	.m3u8	H.264 HEVC (H.265)	AAC

CONTAINER	MEDIA TYPE	EXTENSIONS	VIDEO CODECS	AUDIO CODECS
Audio Video Interleave	AVI	.avi, .divx, .xvid	Uncompressed DivX/Xvid DV/DVCPRO	Dolby® Digital Dolby® Digital Plus™ Dolby® E frames carried in PCM streams MPEG Audio PCM
Adobe® Flash®	F4V	.f4v, .flv	Flash® 9 File H.263 H.264	AAC
Interoperable Mastering Format	IMF	.xml	JPEG2000	PCM
Matroska	MKV	.mkv	H.264 MPEG-2 MPEG-4 part 2 VC-1	AAC Dolby® Digital Dolby® Digital Plus™ WMA, WMA2
MPEG Transport Streams	MPEG TS	.m2ts, .m2t, .mts, .ts, .trp, .mpeg	H.264 HEVC (H.265) MPEG-2 VC-1	AAC AIFF Dolby® Digital Dolby® Digital Plus™ Dolby® E frames carried in PCM streams MPEG Audio PCM WMA, WMA2
MPEG-1 System Streams	MPEG SS	.mpg, .mpeg	MPEG-1 MPEG-2	AAC AIFF Dolby® Digital Dolby® Digital Plus™ MPEG Audio PCM
MPEG-4	MPEG-4	.mp4, .m4v, .f4v	Uncompressed AVC Intra 50/100 DivX/Xvid H.261 H.262 H.263 H.264 JPEG 2000 MJPEG MPEG-2 MPEG-4 part 2 VC-1	AAC Dolby® Digital Dolby® Digital Plus™ PCM WMA, WMA2
MXF	MXF	.mxf	Uncompressed AVC Intra 50/100 VC-3 DV/DVCPRO DV25 DV50 DVCPro HD H.264 JPEG 2000 MPEG-2 Panasonic P2 SonyXDCam, SonyXDCam MPEG-4 Proxy JPEG2000	AAC AIFF Dolby® E frames carried in PCM streams MPEG Audio PCM

CONTAINER	MEDIA TYPE	EXTENSIONS	VIDEO CODECS	AUDIO CODECS
QuickTime®		.mov	Uncompressed Apple® ProRes AVC Intra 50/100 DivX/Xvid DV/DVCPRO H.261 H.262 H.263 H.264 JPEG 2000 MJPEG MPEG-2 MPEG-4 part 2	AAC
Video Object Files	VOB	.vob	MPEG-2	AAC MPEG Audio PCM
WMV/ASF	WMV/ASF	.wmv, .asf	VC-1	WMA, WMA2

CONTAINERS AND CODECS FOR FILE OUTPUT

CONTAINER	VIDEO CODECS	AUDIO CODECS
Raw (No container)	Frame Capture (MJPEG) H.264 HEVC (H.265) MJPEG MPEG2 YUV (uncompressed)	AAC AIFF Dolby® Digital Dolby® Digital Plus™ DTS Express™ MPEG Audio WAV
Apple® HTTP Live Streaming	H.264 HEVC (H.265)	AAC Dolby® Digital Dolby® Digital Plus™
3GPP	H.264	AAC
HDS	H.264	AAC
ISMV for MSS	H.264	AAC Dolby® Digital Dolby® Digital Plus™
MPEG DASH ISO	H.264 HEVC (H.265)	AAC Dolby® Digital Dolby® Digital Plus™
MPEG-2 Transport Stream	H.264 HEVC (H.265) MPEG2	AAC Dolby® Digital Dolby® Digital Plus™ MPEG Audio
MPEG-4	H.264 HEVC (H.265)	AAC Dolby® Digital Dolby® Digital Plus™ DTS Express™
MPEG-4 Flash®	H.264	AAC
QuickTime®	H.264 MPEG2 Apple® ProRes YUV (uncompressed)	AAC AIFF Dolby® Digital Dolby® Digital Plus™ WAV

CONTAINER	VIDEO CODECS	AUDIO CODECS
Ultraviolet	H.264	AAC Dolby® Digital Dolby® Digital Plus™ DTS Express™
MXF	MPEG2	WAV

AUDIO CODEC PASS-THROUGH SUPPORT

Pass-through is supported with Dolby E frames carried in PCM streams.

To obtain this pass-through, specify Uncompressed AIFF or Uncompressed WAV as the output codec. The AWS Elemental software will detect Dolby E frames carried in PCM frames and will pass-through rather than decode the input audio.

Pass-through of Dolby E frames carried in PCM streams does not require the AWS Elemental Audio Decode Package license option.

For pass-through the following parameters are supported:

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

SUPPORTED HLS PLAYER VERSIONS

Generally, all AWS Elemental Conductor File features work with version 2 or above of an HLS player. This table lists features that require a higher player version.

The first column specifies the feature; the second column specifies the feature setup that requires a higher version and the setup that works on version 2; the third column specifies the version.

FEATURE	DESCRIPTION	REQUIRES THIS EXT-X-VERSION OR HIGHER
Integer Durations	HLS output group > Advanced > Floating Point Manifest = checked	3
	HLS output group > Advanced > Floating Point Manifest = unchecked	2
Sample AES Encryption	HLS output group > Advanced > Alternate Manifest Destination > Encryption = SAMPLE-AES	5
	HLS output group > Advanced > Alternate Manifest Destination > Encryption = value other than SAMPLE-AES	2
Audio-only stream with an alternate audio	Use an input file with multiple audio tracks and create two audio selectors: track 1 and track 2 (Add Input button at top of profile) In HLS Output, create one stream with audio+video, and create another with only audio Associate one stream with one HLS output and associate the other stream with a second HLS output. On the audio-only HLS output > Advanced > Alternate audio track = checked.	4
	On the audio-only HLS output > Advanced > Alternate audio track = unchecked.	2
Emit single file	HLS output group > Advanced > VOD Mode = checked. Then Emit Single File field appears. Emit Single File field = checked	4
	Emit Single File field = checked.	2
I-frame only manifest	HLS Output group > Output > Add I-frame Only Manifest = checked	4
	HLS Output group > Output > Add I-frame Only Manifest = unchecked	2

FEATURE	DESCRIPTION	REQUIRES THIS EXT-X-VERSION OR HIGHER
Sample-based encryption with Key format and Key format versions attributes	HLS output group > Advanced > Alternate Manifest Destination > Encryption = value other than 1Mainstream or pHLS	5
	HLS output group > Advanced > Alternate Manifest Destination > Encryption = 1Mainstream or pHLS	2

SUPPORTED CAPTION FORMATS

The tables on the following pages combine information about the input container and captions and output containers. To use this information, find the table that corresponds to the type of output you are producing. Within each table, find the container (first column) and caption format (second column) of the original input. Then in the third column, find the caption formats that can be produced.

For more information, see Setting Up Captions with AWS Elemental Server in the public online documentation.

ADOBE HDS, DASH ISO, MICROSOFT SMOOTH OUTPUT - SUPPORTED OUTPUT CAPTION FORMATS

ORIGINAL INPUT CONTAINER	ORIGINAL CAPTION FORMAT	SUPPORTED OUTPUT CAPTION FORMATS
HLS Container in File Input	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded SMPTE-TT TTML
	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded SMPTE-TT TTML
IMF CPL in File Input	IMSC1 text profile	Burn-in TTML
MP4 Container in File Input	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded SMPTE-TT TTML
	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded SMPTE-TT TTML
MPEG Transport Streams in File Input	DVB-Sub	Burn-in
	SCTE-27	Burn-in DVB-Sub
	Teletext	Burn-in SMPTE-TT TTML
MXF Container in File Input	Ancillary Data	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded SMPTE-TT TTML

	Embedded	Burn-in Embedded Embedded+SCTE-20 RTMP CaptionInfo RTMP CuePoint SCTE-20+Embedded
QuickTime® Container in File Input	Ancillary Data	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded SMPTE-TT TTML
	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded SMPTE-TT TTML
	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded SMPTE-TT TTML
Raw (No Container) in File Input	SCC	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded SMPTE-TT TTML
	SMI	Burn-in SMPTE-TT TTML
	SMTPE-TT	Burn-in SMPTE-TT TTML
	SRT	Burn-in SMPTE-TT TTML
	STL	Burn-in SMPTE-TT TTML
	TTML	Burn-in SMPTE-TT TTML
Transport Stream in File Input	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded SMPTE-TT TTML
	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded SMPTE-TT TTML

APPLE® HLS OUTPUT SUPPORTED OUTPUT CAPTION FORMATS

ORIGINAL INPUT CONTAINER	ORIGINAL CAPTION FORMAT	SUPPORTED OUTPUT CAPTION FORMATS
HLS Container in File Input	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded Web VTT
	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded Web VTT
IMF CPL in File Input	IMSC1 text profile	Burn-in Web VTT
MP4 Container in File Input	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded Web VTT
	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded Web VTT
MPEG Transport Streams in File Input	DVB-Sub	Burn-in
	SCTE-27	Burn-in DVB-Sub
	Teletext	Burn-in Web VTT
MXF Container in File Input	Ancillary Data	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded Web VTT
	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded Web VTT
QuickTime® Container in File Input	Ancillary Data	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded Web VTT
	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded Web VTT
	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded Web VTT

Raw (No Container) in File Input	SCC	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded Web VTT
	SMI	Burn-in Web VTT
	SMTPE-TT	Burn-in Web VTT
	SRT	Burn-in Web VTT
	STL	Burn-in Web VTT
	TTML	Burn-in Web VTT
Transport Stream in File Input	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded Web VTT
	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded Web VTT

MP4 OR 3GPP OUTPUT - SUPPORTED OUTPUT CAPTION FORMATS

ORIGINAL INPUT CONTAINER	ORIGINAL CAPTION FORMAT	SUPPORTED OUTPUT CAPTION FORMATS
HLS Container in File Input	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
IMF CPL in File Input	IMSC1 text profile	Burn-in
MP4 Container in File Input	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
MPEG Transport Streams in File Input	DVB-Sub	Burn-in
	SCTE-27	Burn-in DVB-Sub
	Teletext	Burn-in
	TTML	Burn-in
MXF Container in File Input	Ancillary Data	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded

	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
QuickTime® Container in File Input	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	Ancillary Data	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
Raw (No Container) in File Input	SCC	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	SMI	Burn-in
	SMTPE-TT	Burn-in
	SRT	Burn-in
	STL	Burn-in
	TTML	Burn-in
Transport Stream in File Input	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded

QUICKTIME® OUTPUT - SUPPORTED OUTPUT CAPTION FORMATS

ORIGINAL INPUT CONTAINER	ORIGINAL CAPTION FORMAT	SUPPORTED OUTPUT CAPTION FORMATS
HLS Container in File Input	Embedded	Burn-in Embedded Embedded+Ancillary Data Embedded+SCTE-20 SCTE-20+Embedded
	SCTE-20	Burn-in Embedded Embedded+Ancillary Data Embedded+SCTE-20 SCTE-20+Embedded
IMF CPL in File Input	IMSC1 text profile	Burn-in
MP4 Container in File Input	Embedded	Burn-in Embedded Embedded+Ancillary Data Embedded+SCTE-20 SCTE-20+Embedded
	SCTE-20	Burn-in Embedded Embedded+Ancillary Data Embedded+SCTE-20 SCTE-20+Embedded

MPEG Transport Streams in File Input	DVB-Sub	Burn-in
	SCTE-27	Burn-in DVB-Sub
	Teletext	Burn-in
MXF Container in File Input	Ancillary Data	Burn-in Embedded Embedded+Ancillary Data Embedded+SCTE-20 SCTE-20+Embedded
	Embedded	Burn-in Embedded Embedded+Ancillary Data Embedded+SCTE-20 SCTE-20+Embedded
QuickTime® Container in File Input	Ancillary Data	Burn-in Embedded Embedded+Ancillary Data Embedded+SCTE-20 SCTE-20+Embedded
	Embedded	Burn-in Embedded Embedded+Ancillary Data Embedded+SCTE-20 SCTE-20+Embedded
	SCTE-20	Burn-in Embedded Embedded+Ancillary Data Embedded+SCTE-20 SCTE-20+Embedded
Raw (No Container) in File Input	SCC	Burn-in Embedded Embedded+Ancillary Data Embedded+SCTE-20 SCTE-20+Embedded
	SMI	Burn-in
	SMTPE-TT	Burn-in
	SRT	Burn-in
	STL	Burn-in
	TTML	Burn-in
Transport Stream in File Input	Embedded	Burn-in Embedded Embedded+Ancillary Data Embedded+SCTE-20 SCTE-20+Embedded
	SCTE-20	Burn-in Embedded Embedded+Ancillary Data Embedded+SCTE-20 SCTE-20+Embedded

RAW (NO CONTAINER) OUTPUT - SUPPORTED OUTPUT CAPTION FORMATS

ORIGINAL INPUT CONTAINER	ORIGINAL CAPTION FORMAT	SUPPORTED OUTPUT CAPTION FORMATS

HLS Container in File Input	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCC SCTE-20+Embedded SMI SMPTE-TT TTML SRT Web VTT
	Embedded	Burn-in Embedded Embedded+SCTE-20 SCC SCTE-20+Embedded SMI SMPTE-TT TTML SRT Web VTT
IMF CPL in File Input	IMSC1 text profile	Burn-in SMI TTML SRT Web VTT
MP4 Container in File Input	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCC SCTE-20+Embedded SMI SMPTE-TT TTML SRT Web VTT
	Embedded	Burn-in Embedded Embedded+SCTE-20 SCC SCTE-20+Embedded SMI SMPTE-TT TTML SRT Web VTT
MPEG Transport Streams in File Input	DVB-Sub	Burn-in SMPTE-TT
	SCTE-27	Burn-in DVB-Sub SMPTE-TT
	Teletext	Burn-in SMI SMPTE-TT TTML SRT Web VTT

MXF Container in File Input	Ancillary Data	Burn-in Embedded Embedded+SCTE-20 SCC SCTE-20+Embedded SMI SMPTE-TT TTML SRT Web VTT
	Embedded	Burn-in Embedded Embedded+SCTE-20 SCC SCTE-20+Embedded SMI SMPTE-TT TTML SRT Web VTT
QuickTime® Container in File Input	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCC SCTE-20+Embedded SMI SMPTE-TT TTML SRT Web VTT
	Embedded	Burn-in Embedded Embedded+SCTE-20 SCC SCTE-20+Embedded SMI SMPTE-TT TTML SRT Web VTT
	Ancillary Data	Burn-in Embedded Embedded+SCTE-20 SCC SCTE-20+Embedded SMI SMPTE-TT TTML SRT Web VTT
Raw (No Container) in File Input	SCC	Burn-in Embedded Embedded+SCTE-20 SCC SCTE-20+Embedded SMI SMPTE-TT TTML SRT Web VTT

	SMI	Burn-in SMI SMPTE-TT TTML SRT Web VTT
	SMTPE-TT	Burn-in SMI SMPTE-TT TTML SRT Web VTT
	SRT	Burn-in SMI SMPTE-TT TTML SRT Web VTT
	STL	Burn-in SMI SMPTE-TT TTML SRT Web VTT
	TTML	Burn-in SMI SMPTE-TT TTML SRT Web VTT
Transport Stream in File Input	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCC SCTE-20+Embedded SMI SMPTE-TT TTML SRT Web VTT
	Embedded	Burn-in Embedded Embedded+SCTE-20 SCC SCTE-20+Embedded SMI SMPTE-TT TTML SRT Web VTT
	DVB-Sub	Burn-in SMPTE-TT
	SCTE-27	Burn-in DVB-Sub SMPTE-TT
	Teletext	Burn-in SMI SMPTE-TT TTML SRT Web VTT

TRANSPORT STREAM ARCHIVE OUTPUT - SUPPORTED OUTPUT CAPTION FORMATS

ORIGINAL INPUT CONTAINER	ORIGINAL CAPTION FORMAT	SUPPORTED OUTPUT CAPTION FORMATS
HLS Container in File Input	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
IMF CPL in File Input	IMSC1 text profile	Burn-in DVB-Sub Teletext
MP4 Container in File Input	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
MPEG Transport Streams in File Input	DVB-Sub	Burn-in DVB-Sub
	SCTE-27	Burn-in DVB-Sub SMPTE-TT
	Teletext	Burn-in DVB-Sub
MXF Container in File Input	Ancillary Data	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded Teletext
QuickTime® Container in File Input	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	Ancillary Data	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
Raw (No Container) in File Input	SCC	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	SMI	Burn-in
	SMTPE-TT	Burn-in

	SRT	Burn-in
	STL	Burn-in
	TTML	Burn-in
Transport Stream in File Input	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded

ULTRAVIOLET FORMAT OUTPUT - SUPPORTED OUTPUT CAPTION FORMATS

ORIGINAL INPUT CONTAINER	ORIGINAL CAPTION FORMAT	SUPPORTED OUTPUT CAPTION FORMATS
HLS Container in File Input	Embedded	Burn-in CFF-TT Embedded Embedded+SCTE-20 SCTE-20+Embedded
	SCTE-20	Burn-in CFF-TT Embedded Embedded+SCTE-20 SCTE-20+Embedded
IMF CPL in File Input	IMSC1 text profile	Burn-in CFF-TT
MP4 Container in File Input	Embedded	Burn-in CFF-TT Embedded Embedded+SCTE-20 SCTE-20+Embedded
	SCTE-20	Burn-in CFF-TT Embedded Embedded+SCTE-20 SCTE-20+Embedded
MPEG Transport Streams in File Input	DVB-Sub	Burn-in
	SCTE-27	Burn-in DVB-Sub
	Teletext	Burn-in CFF-TT
MXF Container in File Input	Ancillary Data	Burn-in CFF-TT Embedded Embedded+SCTE-20 SCTE-20+Embedded
	Embedded	Burn-in CFF-TT Embedded Embedded+SCTE-20 SCTE-20+Embedded
QuickTime® Container in File Input	Ancillary Data	Burn-in CFF-TT Embedded Embedded+SCTE-20 SCTE-20+Embedded

	Embedded	Burn-in CFF-TT Embedded Embedded+SCTE-20 SCTE-20+Embedded
	SCTE-20	Burn-in CFF-TT Embedded Embedded+SCTE-20 SCTE-20+Embedded
Raw (No Container) in File Input	SCC	Burn-in CFF-TT Embedded Embedded+SCTE-20 SCTE-20+Embedded
	SMI	Burn-in CFF-TT
	SMTPE-TT	Burn-in CFF-TT
	SRT	Burn-in CFF-TT
	STL	Burn-in CFF-TT
	TTML	Burn-in CFF-TT
SCTE-20	Burn-in CFF-TT Embedded Embedded+SCTE-20 SCTE-20+Embedded	

XDCAM OUTPUT SUPPORTED OUTPUT CAPTION FORMATS

ORIGINAL INPUT CONTAINER	ORIGINAL CAPTION FORMAT	SUPPORTED OUTPUT CAPTION FORMATS
HLS Container in File Input	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
IMF CPL in File Input	IMSC1 text profile	Burn-in
MP4 Container in File Input	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
MPEG Transport Stream in File Input	DVB-Sub	Burn-in DVB-Sub
	SCTE-27	Burn-in
	Teletext	Burn-in Teletext

MXF Container in File Input	Ancillary Data	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
QuickTime Container in File Input	Ancillary Data	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
Raw (No container) in File Input	SCC	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	SMI	Burn-in
	SMPTE-TT	Burn-in
	SRT	Burn-in
	STL	Burn-in
	TTML	Burn-in
Transport Stream in File Input	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded