ELEMENTAL® LIVE

API AND USER GUIDE

Applicable to 2.26 and higher RELEASES

Feb 2024

TABLE OF CONTENTS

OVERVIEW

PURPOSE

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

PRODUCT OVERVIEW

Elemental Live is a powerful media encoder which can accept input through an HD-SDI interface, IP over Ethernet, or a local file. It can produce multiple video output streams using a variety of live streaming protocols.

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

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

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

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

The SNMP interface allows basic monitoring and control of the Elemental Live system. It allows a management system to query the state of the service and Live Events, as well as start and stop Live Events.

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

LIVE EVENTS

A Live Event is an encoding session started and stopped by the operator of the system. Live Events are created by selecting an input from the available HD-SDI capture devices or specifying a network stream location. The operator can also select a file input, which will be decoded in real-time to simulate a live input. This feature is useful for testing encoding parameters and output destinations when a live source is not available.

When using the web interface, the user can click the "Preview" button once they have filled in input parameters. Elemental Live will attempt to acquire the input source and display a scaled frame and audio level for the user to verify their source is correctly connected to Elemental Live.

The operator then defines the output stream encoding parameters that they would like the system to produce. See Supported Codecs for information about which codecs Elemental Live supports. The user can select which GPU they would like the stream to utilize, or allow the system to automatically select the GPU.

Output streams can be connected to multiple output groups, which define endpoints for streaming applications. Elemental Live currently supports Adobe RTMP outputs for Adobe Flash Media Server, push encoding for Microsoft Smooth Streaming, Apple iPhone live streaming, Dynamic Adaptive Streaming over HTTP (DASH), UDP output, reliable transport stream output, and archiving to the local hard drive. See Supported Codecs for a list of all the supported container formats.

LIVE EVENT PROFILES

A Live Event Profile is a saved Live Event definition that includes all output settings for a Live Event and, optionally, the input settings. Live Events can be submitted with a Live Event Profile ID and input parameters to re-use previously entered settings. Note that if a Live Event Profile is edited, those changes are only applied to Live Events created after the change. Live Events already in progress or in the PENDING state will retain the settings with which they were submitted.

Some example Live Event Profiles are supplied by default in each release of the Elemental Live 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.

PRESETS

A preset is a predefined group of settings for a single output stream. A preset allows the user to create output streams targeted at a particular device or standard output format. For example, the H.264 HD Medium preset produces a stream with H.264 video at 1280 x 720 resolution and 2.4 mbit/s, and AAC audio at 128 kbit/s. Elemental maintains a list of common presets that are delivered to the Elemental Live system via software updates. Additionally, the user can specify named presets using any of the interfaces to the Elemental Live system.

SCHEDULES

Schedules can be created to run certain Live Event Profiles at scheduled times, or a set of repeating times. For example, a schedule can be created to run every weekday from 1:00PM to 2:00PM to stream a program. Each of these scheduled programs will be converted to a Live Event.

MPTS MULTIPLEXER

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

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

If these criteria are met, the LiveEvent will be available for inclusion. Live Events can be added and removed from the MPTS at any time, but must be removed from one MPTS before joining another.

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

NOTIFICATION

Users can set up a Live Event so that a notification is sent if the Live Event is started, stopped, or has an alert or error. 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 Live Event

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

STATISTICS

Elemental Live 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 Live Event is run. Examples of these operations include:

- Running custom validation on input or output files before or after a conversion
- Running custom notifications before or after the Live Event is run

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

CUSTOM SCRIPTS

For each Live Event 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 Live Event UI 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 live_runner.output log file.

The pre processing script is called from the elemental_se service just before the Live Event runs and must have execute permission for the elemental user. The Live Event's state is changed to PREPROCESSING when the pre script is running, and POSTPROCESSING when the post script is running. The Live Event can still be cancelled when it is in one of these states. The reported start and end times for a Live Event 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** ID of the Live Event
- **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,network_input,smpte2022_dash7_network_input,smpte2110_input,device_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 (Archive, Apple HLS, 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 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 Live Event. Only one message can be sent back to the system and stored with the Live Event in the database; however, all messages and outputs to STDOUT will be present in the sequencer log file. Errors will not allow the Live Event 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. Elemental recommends implementing scripts in a language that includes JSON parsing libraries.

TROUBLESHOOTING

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

For additional support, contact your Elemental support representative, or use the AWS Elemental Support

Center in the AWS Console - http://amzn.to/AWSElementalSC -

WEB INTERFACE QUICK START GUIDE

WEB INTERFACE QUICK START GUIDE

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

DEFINITION OF COMMON TERMS

- **Live Event:** a stream or streams to be broadcast in real-time, specifies input source(s), output stream(s), output group(s), optionally may include a start/stop time and video effects to be applied to the output streams.

  How to create a Live Event

- **Preset:** A preset is a predefined group of settings for a stream. 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 Live Event

  Note about editing Presets

- **Live Event Profile:** A Live Event Profile is a saved Live Event definition that includes all output settings and may optionally include input details. Live Event Profiles provide a quick method for creating identical Live Events.

  Using the Live Event Profiles Page to Create a New Live Event

  How to create a Live Event Profile

- **Input:** An input contains information about the source input(s) for the broadcast streams. The input can be from an HD-SDI input card, from an RTMP, UDP or RTP multicast or unicast network stream, or from a file. A Live Event can define multiple inputs, either for input failover or for dynamic input switching.

  Switching between inputs

- **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 CDN information or delivery addresses for streaming groups (Adobe RTMP, Smooth Streaming) or file names and network locations for archive groups. In addition, each stream may need specific information for each group, such as the file name for the output or the individual IP address target.

  How to set up an Archive Group with Outputs

  How to set up an Apple HLS Group with Outputs

  How to set up an MS Smooth Streaming Group with Outputs

  How to set up an Adobe RTMP Group with Outputs

  How to set up a UDP/TS Group with Outputs

- **Output:** An output is made up of the combination of a stream and a group.
- **Timing:** A timing object directs a Live Event to start and stop at a particular time.
- **Schedule:** A schedule contains information for scheduling repeating Live Events from a particular Live Event Profile.

  How to create a Schedule

- **MPTS:** An MPTS represents a Multi-Program Transport Stream multiplexer.

  How to set up an MPTS

- **Preset Category:** User-defined category that can be used to organize presets.

DEFINITION OF COMMON ICONS

On the Event Control page, as well as the Presets and Live Event Profiles and many other pages, icons are used to indicate both state and available actions. Many of these icons are not explicitly labeled (though if you hover over the icon, a tooltip will appear to indicate the icon's action). An example of how this looks can be found in the Event Control page screenshot. Clicking on an icon will trigger the action associated with that icon.

Show:

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

Edit:

This icon indicates that the given object is allowed to be edited. Note that Live Events may not be edited while they are running and the default Presets that come loaded with Elemental Live cannot be edited.

Duplicate:

This icon indicates the duplication of an object. It is used on the Event Control page, the Presets page and the Profiles page. Clicking this icon will navigate to the New page for the given object, with all of the information filled out from the duplicated object.

Create Live Event:

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

Delete:

This icon allows for the deletion of objects. Note that the collection of default Presets that come loaded with Elemental Live cannot be deleted.

Cancel/Stop:

This icon indicates two separate actions. In the last icon column this icon is found when a Live Event is pending, and it is used to cancel the Live Event. In the second-to-last icon column this icon is found when the Live Event is running, and in this position it indicates a command to stop the Live Event.

☐ Reset:
This icon is used to reset a cancelled, completed or errored Live Event.

☐ Play:
This icon is used to start a pending Live Event. It also appears beside each Input when viewing a currently running Live Event. Here it can be used to switch to an Input on the fly during the course of the Live Event.

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

There are six base pages for the default web interface

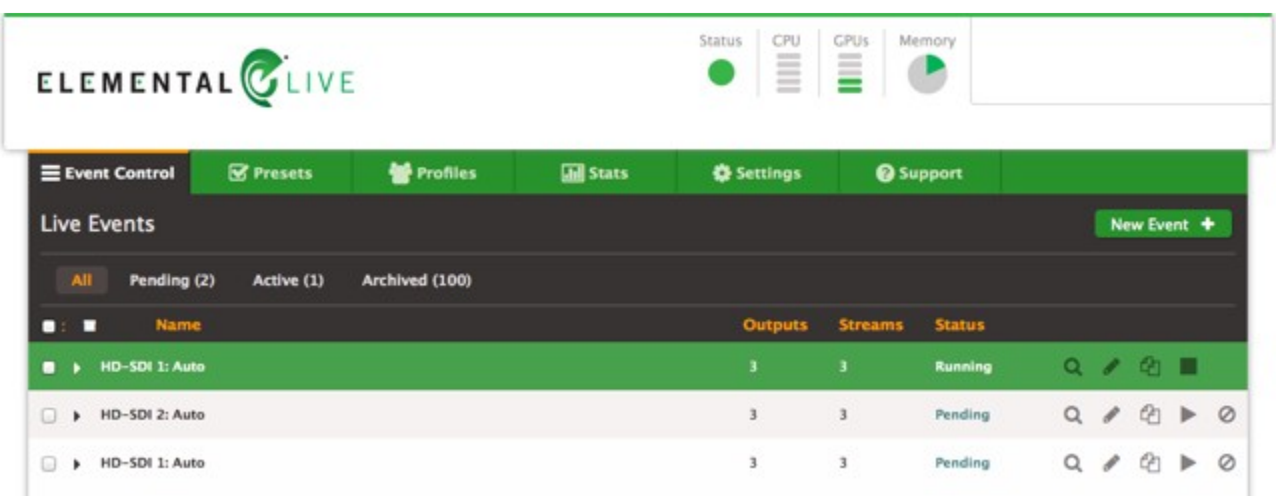
- Event Control: View status of current Live Events, current and past Live Event details, or create new Live Events
- Presets: View, create and edit Presets and Preset Categories
- Live Event Profiles: View, create and edit Live Event Profiles
- Schedules: View, create and edit Schedules
- Stats: Provides statistics for Elemental Live
- Alerts: Access system and Live Event alerts
- Settings: Modify Elemental Live settings
- Support: Documentation for the web interface, the REST interface, and the SNMP API

TYPICAL STEPS FOR GETTING STARTED WITH ELEMENTAL LIVE

Point a web browser at the Elemental Live web address:

https://<ip address of server>

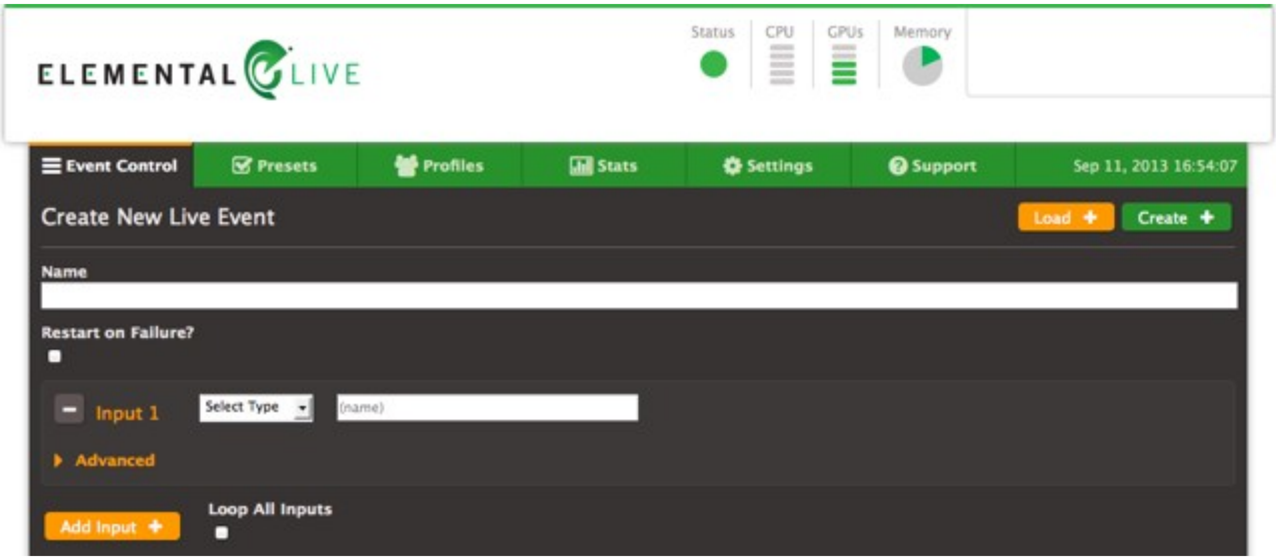
You should see a screen like this:



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

CREATING A NEW LIVE EVENT

After selecting the *Event Control* page, simply click the "New Event" button.



To begin, give the Live Event a name, and set any other options you would like this Live Event to use.

CONFIGURING INPUTS

To configure an input, select an input source (HD-SDI, Network (UDP, RTP, RTMP), or File) and enter any fields such as a network or file location that the input source requires. Additional inputs can be added by clicking the "Add Input" button. When more than one input is present, the input order can be adjusted using the orange up and down buttons.

Beside each input, the "Preview" button can be used to verify that the Elemental Live system can access your source correctly. The preview window will display a sample video frame from the source and a summary of its metadata such as programs, audio, and video streams.

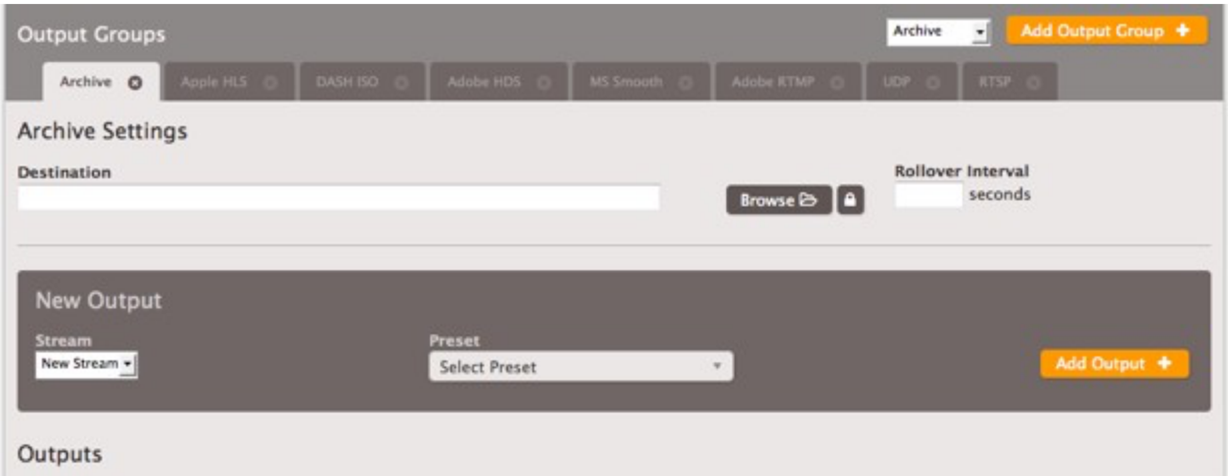
CONFIGURING OUTPUTS

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 Live Event 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 Live Event (Archive, Apple HLS, MS Smooth, Adobe RTMP, or UDP). 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 Live Event is saved.

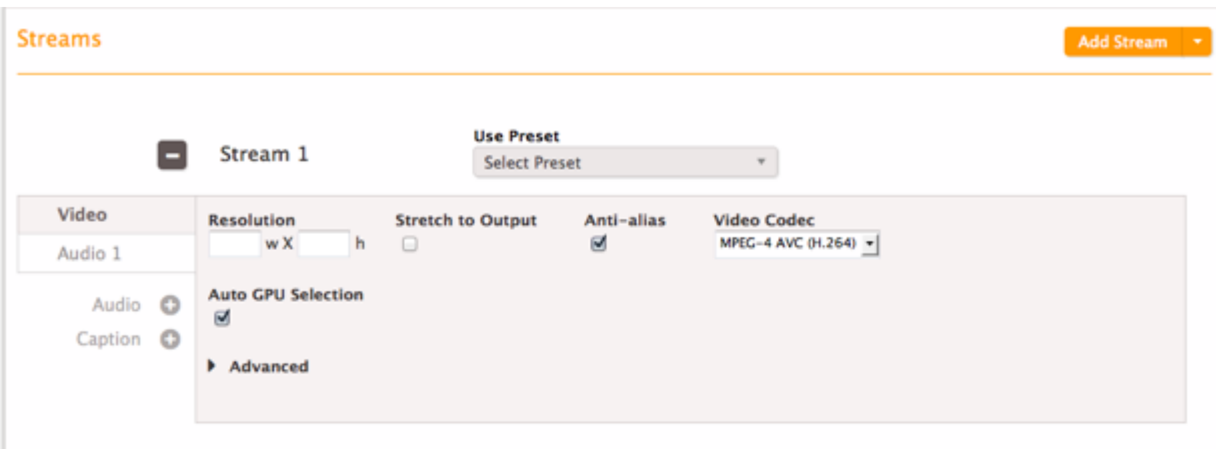
Outputs should be added to this group in the New Output box by selecting 'New Stream' or an existing stream (to re-use video and audio encoding parameters). A Preset can be applied to a new output being added when 'New Stream' is selected. Selecting a Preset from within the New Output box applies the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

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



CONFIGURING STREAMS

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



SETTING UP AN ARCHIVE GROUP

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

The names of outputs in an Archive 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 the filename will depend purely on the individual output's name modifier.

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. If no base file name is selected, then the filename will simply be the name modifier. 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 Apple Live Group **Destination** field accepts either a local output directory or an HTTP endpoint.

- When creating files on disk, the Apple Live Group uses the same file naming mechanism as the Archive Group. The destination directory is also where the .m3u8 files that contain the stream information will be created. Unlike the Archive Group, Apple Live Groups require that a base file name be set. If clients will view streams from this location, the directory should be directly servable by a web server (i.e. publicly accessible).
- Pushing to an HTTP endpoint also requires the specification of a base file name. For example: *http://<server_name>/directory/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 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 **Publish Point** field specifies the output server location and stream name that will match the Smooth Streaming Publishing Points that have been configured on the IIS server. For example, a Publish Point would be *http://iis/livesmooth/livesmoothstream1.isml*. 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.

SETTING UP AN ADOBE RTMP GROUP

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

The **RTMP Endpoint** is used by the CDN to specify the ingest point. An example would be: *rtmp://p.ep9999.i.akamaientrypoint.net/EntryPoint*. Clicking the credentials button next to the RTMP Endpoint displays **Username** and **Password** fields that are used for CDNs such as Akamai, which require authentication.

SETTING UP A UDP/TS GROUP

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

The MPTS Membership defines whether the output should be a member of an MPTS. If â€˜Noneâ€™ or â€˜Remoteâ€™ is selected, the URI specifies the address where the UDP stream is to be published. A port must be specified for a stream.

- Unicast: use the URI field to specify the IP address of the device you wish to stream to: *udp://<ip_addr>:<port>*. An example of a valid URI entry would be *udp://10.1.1.10:5020*.

- Multicast: use the URI field to specify the IP address of the device you wish to stream to with the additional TTL parameter: *udp://<ip_addr>:<port>?ttl=x*. An example of a valid URI entry would be *udp://239.255.11:5020?ttl=1*.

CREATING A LIVE EVENT 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:

- Live Event XML
- Live Event Profile

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

Selecting Live Event Profile will display a drop-down list of all available Profiles. Applying a profile from the list will reload the Live Event 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 LIVE EVENT

After all of your Output Groups, Streams and Outputs have been set up, click the "Create" button in the upper right to create the Live Event. The Live Event will then be placed in a "Pending" state. To start the Live Event, click the "Start" button.

The main view for a Live Event is called the **Control Panel**. From the "Control Panel" you can monitor a running Live Event, see a preview window with live updated frame captures, see and switch the currently running input, pause and unpause outputs and whole output groups, stop and reset the Live Event, and more. If you need more complete information about the Live Event click "Details" near the top of this page. The **Details** page contains the complete set of Live Event parameters for reference.

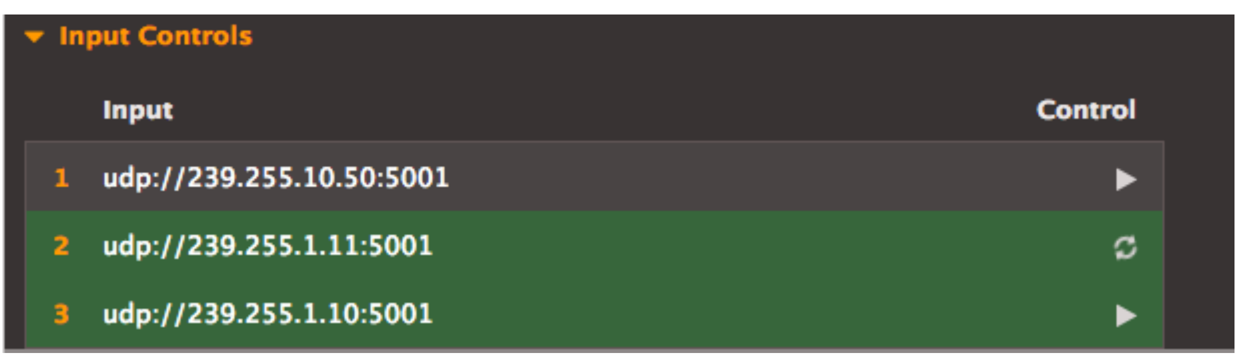
SWITCHING INPUTS

A Live Event can contain multiple input sources, which get transcoded sequentially by default. While inputs can be switched manually via the Input Controls of the Control Panel, there are also several tools in place to manage the automatic switching of inputs.

Failover Conditions dictate when inputs 'fail over' to the next subsequent input, due to unexpected input failures. Multiple failover conditions can be set, but only one condition needs to be met for an input to failover.

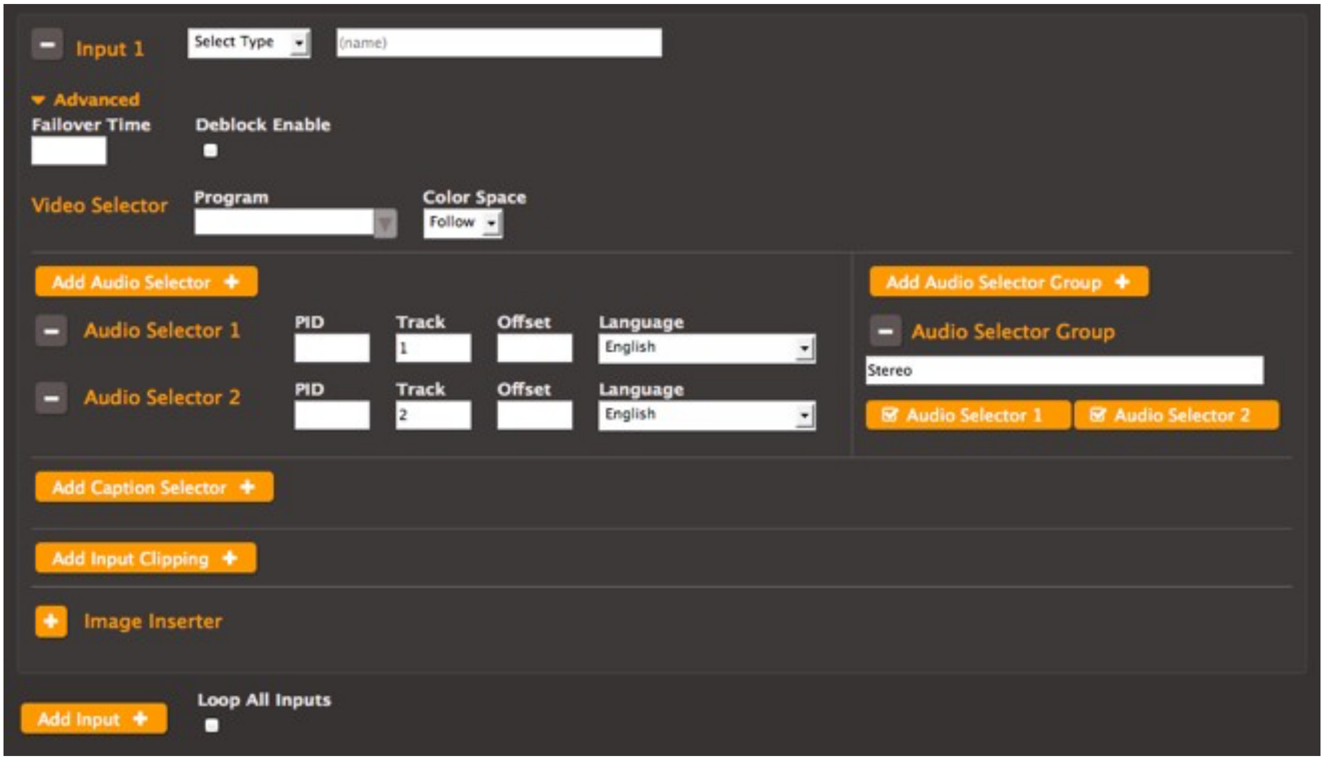
Hot Backup pairs an input with the next or previous listed input to provide simultaneous demuxing and decoding of both inputs. This allows instantaneous failover between inputs, and automatic failback to a restored input. Both Inputs are demuxed and decoded, and only one of them is connected to the rest of the encoding pipeline. Each input can then have its own set of failover conditions. If a failover condition on the first sequential input is met, the system fails over to the subsequent paired input immediately (as it is already being decoded). If there are no failover conditions set, the Hot Backup pair will decode simultaneously with failover a completely manual process.

When a pair of inputs have Hot Backup enabled, they may also have unique Failback Rules specified. Failback Rules dictate the conditions to restore an input to an active state, based on a specified error free interval. With Hot Backup and Failover rules defined, valid output is guaranteed, as long as only one of the inputs is corrupted at a given time.



ADVANCED AUDIO TRACK SELECTIONS

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



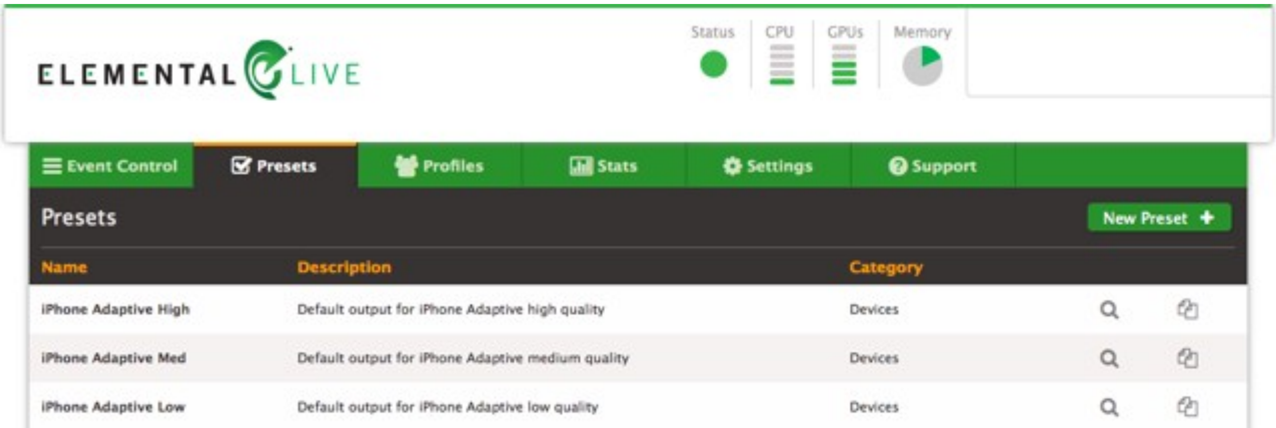
In the Audio settings of the output, note that your Audio Selectors and the "Stereo" group are available for this output.

For more information, see Audio Selector

USING PRESETS

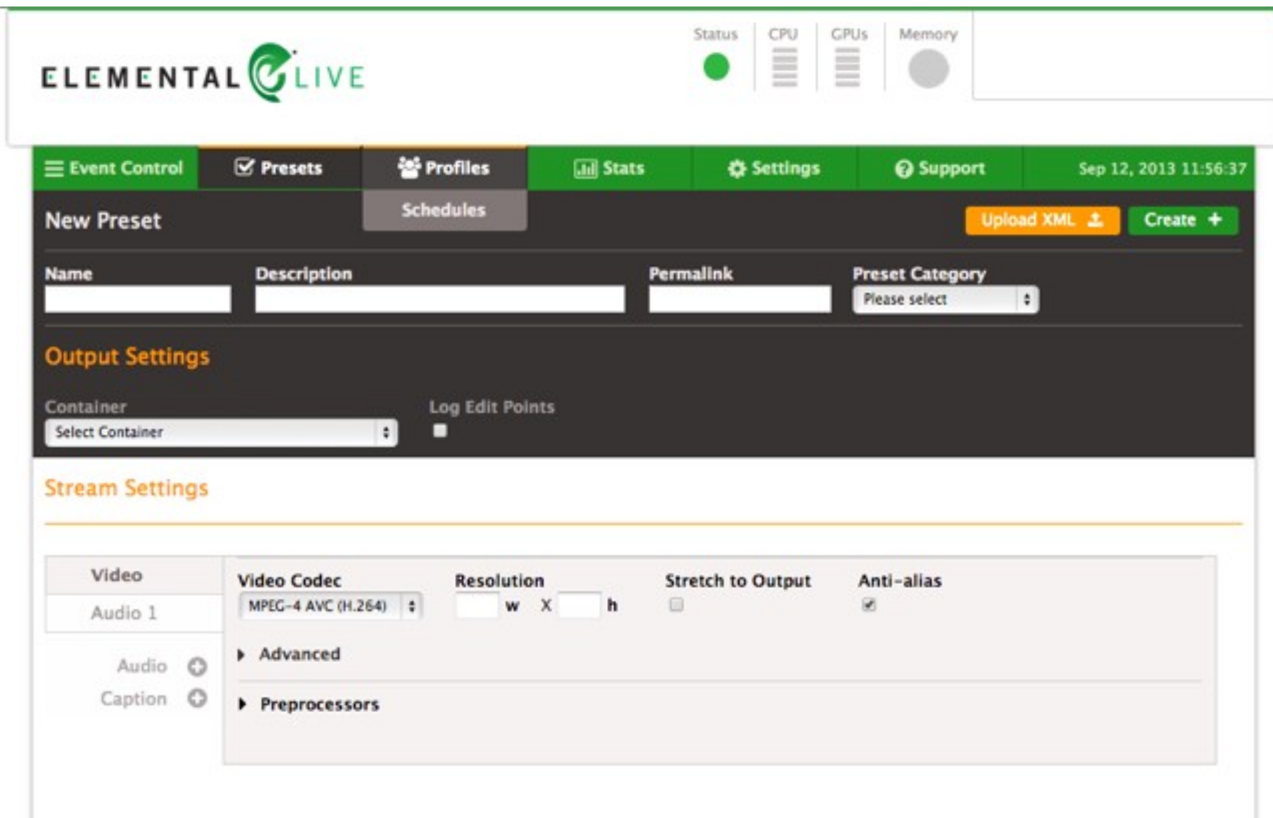
Presets simplify stream creation when the same encoding parameters will be used repeatedly.

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



CREATING A NEW PRESET

Duplicate an existing Preset and modify it to meet your target settings or click the *New Preset* button. For example, duplicating the iPad Adaptive 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 LIVE EVENT

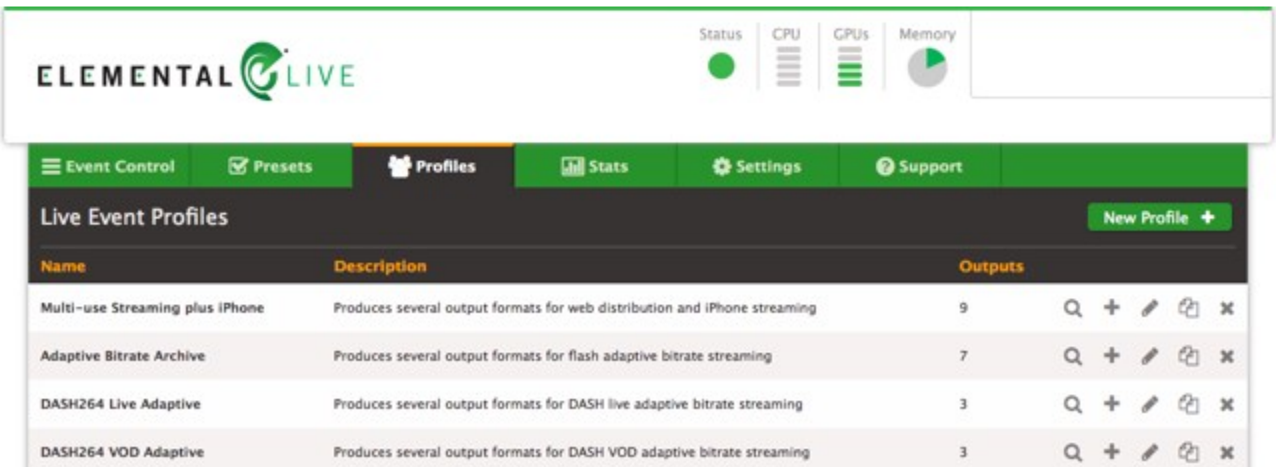
Creating a Preset using the Stream settings of an existing Live Event is useful if these settings will be used again. Navigate to the "Show" page for the Live Event and find the Stream that you wish to save as a preset. Next to the Stream label is a "Save as Preset" button. Clicking this button brings 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 Live Event Profiles and Live Events that had been created using this Preset will not be updated. If a Preset must be edited, all associated Live Event Profiles will need to be updated to use the updated Preset.

USING LIVE EVENT PROFILES

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



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

CREATE A NEW LIVE EVENT FROM A LIVE EVENT PROFILE

From the New Live Event page: Click "Load" (next to the green create button). Select "Live Event Profile" from the drop down menu that appears below the button. Choose the desired profile from the list and click "Apply" to load the settings. **This action will reload the page and overwrite all fields.**

From the main Profiles page: There is a Create Live Event icon listed for each Live Event Profile on the main Profiles page. Clicking this icon will take you to the New Live Event page with all the information from the given Live Event Profile already filled out. Additionally, any changes to the Live Event Profile's parameters can be made at this time as well.

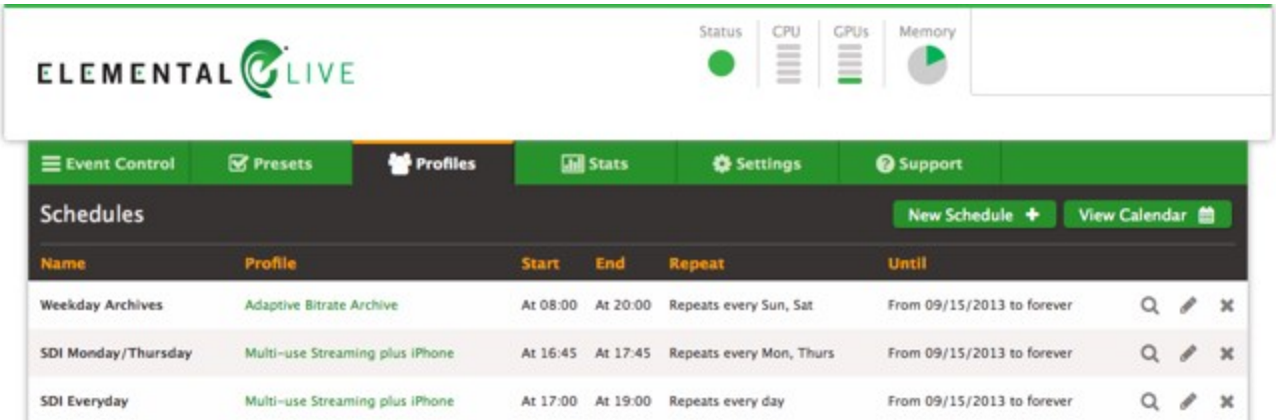
CREATING A NEW LIVE EVENT PROFILE

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

The top section of the detailed Live Event Profile page shows the information related to the Live Event Profile. The lower section of the page shows the streams and groups associated with the Live Event Profile. More details on the available Live Event Profile Parameters can be found on the Parameters page.

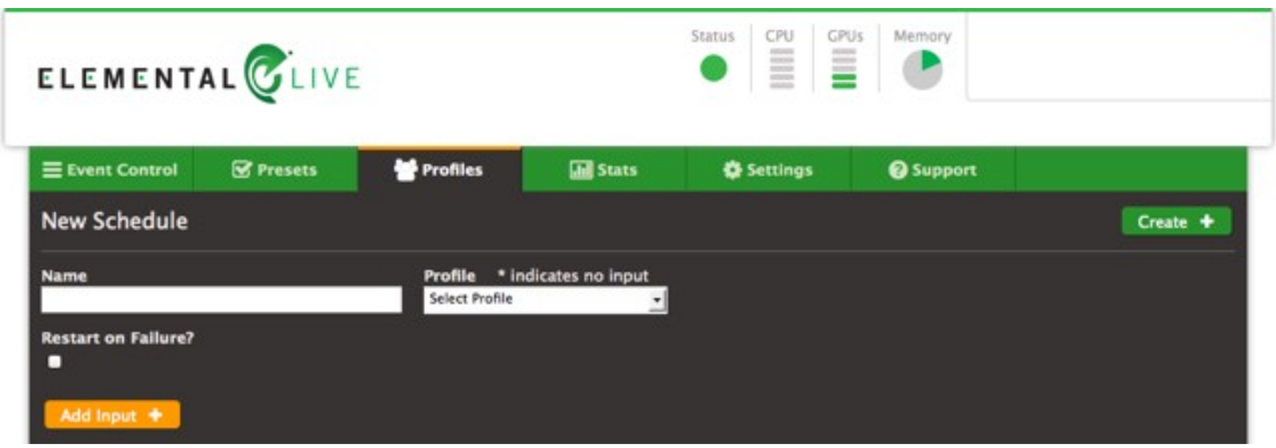
USING SCHEDULES

If you have Live Events that will be repeated at specific intervals, you can set up a Schedule to automatically generate those Live Events. The *Schedules* page displays all of the scheduled Live Events.



CREATING A NEW SCHEDULE

A Schedule is created using a pre-existing Live Event Profile to generate the Live Event output settings, so first make sure that a Live Event Profile with the desired output settings exists. This Live Event Profile may optionally indicate an input method, but this is not required. Clicking *New Schedule* on the Schedules page will bring you to the New Schedule page.



On the New Schedule page select the desired Live Event Profile, Node, and any desired failure settings. If the Live Event Profile does not have an input specified, or if a different input is desired for the scheduled Live Events, click the *Add Input* button and fill out the desired input settings.

The right side of the screen contains the information encapsulating the repeating schedule of the Live Events. First, fill out the desired Start Time and End Time of the first scheduled Live Event. The date specified in the Start Time field represents the date of the first repeating Live Event. Next specify how long this Schedule is to be repeated, either forever or until a specific date. Then specify the repeating schedule. This can be specified on a daily basis (i.e. every 2 days), or on a weekly basis (i.e. every M, W, F). There are also options for repeating on a monthly basis, either on a specific day (i.e. the 15th of the month) or a specific day of the week (i.e. the Second Thursday of the month).

Once a Schedule has been created, the scheduled Live Events for the next three months will be displayed on the calendar. The Live Events themselves are only created 24 hours in advance. When the actual Live Event has yet to be created, then the scheduled Live Event will be displayed as a light blue color and clicking on the Live Event will navigate to the Schedule page. When the Live Event has been created it will show up in the Live Events list, and the Live Event will be displayed in dark green on the calendar. Clicking on the Live Event will navigate to the Show Live Event page.

USING MPTS MULTIPLEXERS

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

CREATING A NEW MPTS MUX

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

ADDING CHANNELS

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

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

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

SETTING UP NETWORK REDUNDANCY FOR ELEMENTAL MPTSES

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

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

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

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

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

For SPTS data:

- Primary Destination
- Secondary Destination

For initial complexity estimates:

- Primary Complexity Transmit/Receipt Destination

- Secondary Complexity Transmit/Receipt Destination

For final allocations:

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

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

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

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

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

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

PID ASSIGNMENT

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

If the Live Event is later configured to use an additional PID, a new PID value will be automatically assigned within the MPTS. Changing an existing PID value, or removing an optional PID from the Live Event will have no effect on the MPTS PID configuration. Once again, these values can be directly edited at the MPTS at any time.

STARTING OR STOPPING THE MPTS

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

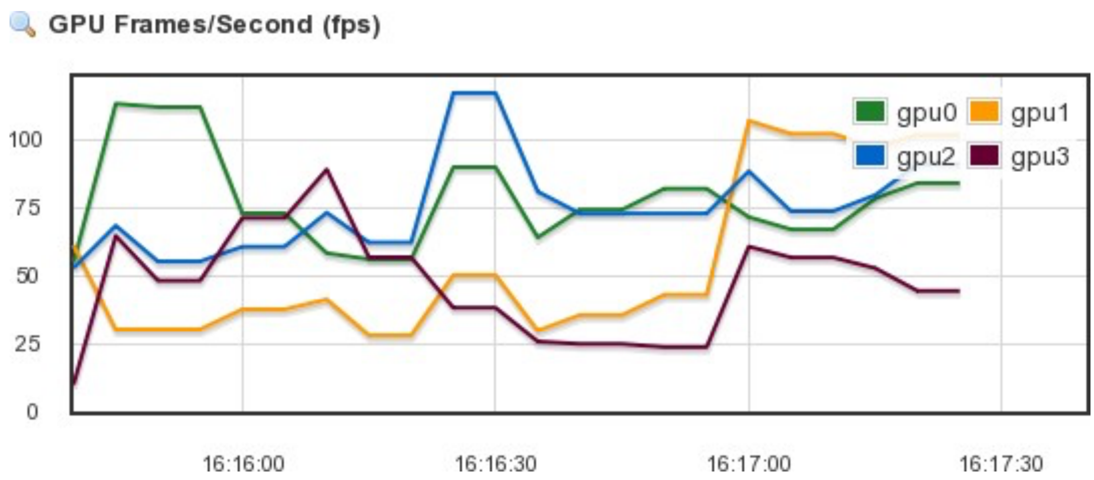
USING THE STATS PAGE

The stats pages provide various statistics about Elemental Live. The Stream Statistics page displays information about the currently running streams, while the Node stats page displays information about the Elemental Live server. In addition, the stats pages provide access to the alerting system as well as a browsable list of log files maintained by the system.

CHARTS

A variety of charts that display 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/Second being transcoded on each GPU is shown below.



STREAM STATISTICS

Stream Statistics can be found by navigating to the main stats page. This includes charts providing information on the Percent Realtime and Total Frames/Second of the currently running Live Events. 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 information on the Elemental Live node itself. This information includes the node's status, the number of Live Events currently running, the number of completed Live Events, and the average output FPS for that node. There are also charts showing the Percent CPU Utilization and the GPU Frames/Second being processed by the node.

In order to access more detailed statistics for the node, click on the node's name to navigate to the Node stats page.

NODE STATISTICS

Detailed information about the Elemental Live node can be found on the Node stats page, including charts providing information on the node's CPU usage, Memory usage, Disk usage, GPU temperature and GPU Frames/Second.

At the bottom of the page is a list of the currently running Live Events as well as the last ten completed Live Events. Links to more detailed information about each Live Event are available.

USING THE ALERTS PAGE

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

Alerts can be generated for system level events including:

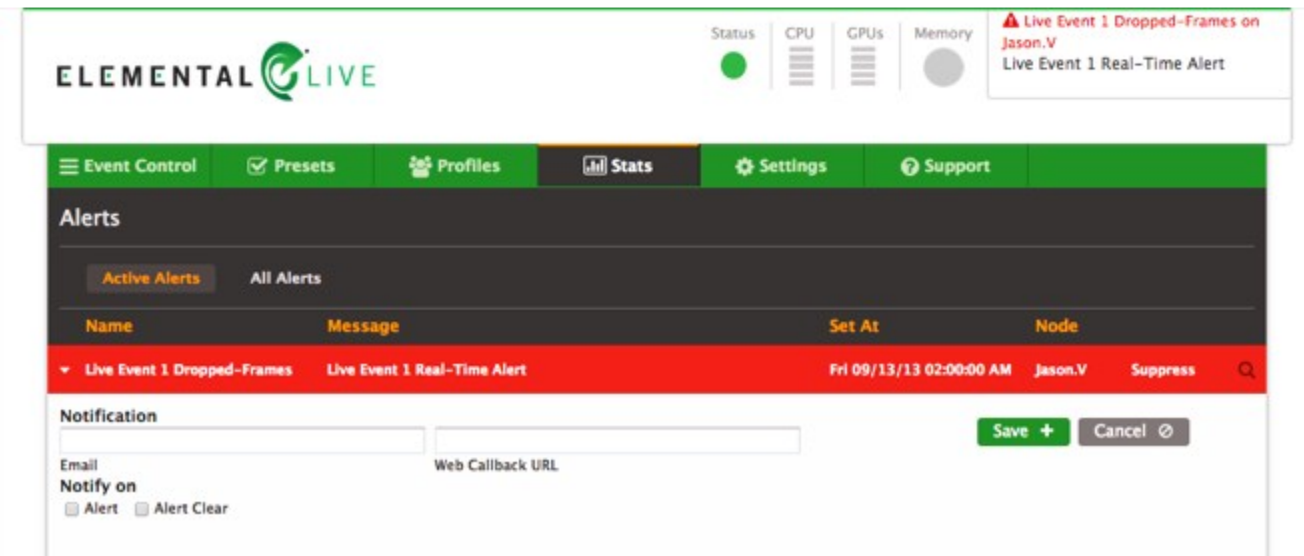
- **CPU Alert** - Cumulative CPU usage is too high
- **Disk Alert** - A disk partition is almost full
- **GPU Temperature Alert** - The GPU temperature is too high

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

Alerts can also be generated for an individual Live Event, such as:

- **Realtime Alert** - The Live Event is running below realtime
- **Network Input Alert** - The Live Event is not receiving data on its network input port
- **RTP Input Alert** - The Live Event lost sync with RTP headers
- **Video Input Alert** - The Live Event is not receiving video data to decode
- **Audio Input Alert** - The Live Event is not receiving audio data to decode
- **Output Alert** - The system cannot send output to the specified endpoint, or has insufficient outgoing throughput
- **HD-SDI Input Alert** - The Live Event is not receiving HD-SDI input
- **Output Lock Alert** - The Live Event failed to lock outputs to external encoder

Notification settings for Live Event alerts can be set on the 'New Live Event' page by checking the "Alert" and "Clear" boxes in the Notification section. These settings can be adjusted on the Alerts page while the Live Event is running; however, those changes will not persist when the Live Event is reset.



USING THE SETTINGS PAGE

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

GENERAL SETTINGS

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

There are also options for managing the cluster-wide background tasks that auto-archive and auto-delete Live Events and delete old thumbnail images off each node. Auto-archive will remove completed Live Events from the displayed Live Events list. Auto-delete permanently removes Live Events 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_Live_2.19.2.0_2014-09-11_05-13-04.tar`, in the backup location `/home/elemental/database_backups`:


```
$ cd /opt/elemental_se/
$ sudo ./configure --restore-db-backup /home/elemental/database_backups/elemental-db-backup_Live_2.19.2.
```

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

NETWORK SETTINGS

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

CURRENT SETTINGS

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

HOSTNAME, DNS, & NTP

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

NETWORK DEVICES

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

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

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

RESTORE DEFAULTS

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

MOUNT POINT SETTINGS

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

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

FIREWALL SETTINGS

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

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

SNMP SETTINGS

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

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

AUTHENTICATION SETTINGS

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

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.

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.

WEB SERVICES REST INTERFACE

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

- REST Basics
  - HTTP Headers
  - API Versions
  - Simple Examples
  - Clean XML
  - Schema Definitions
  - Errors and Warnings
- Live Events
  - Example XML: Create a Live Event from a Live Event Profile
  - Example XML: Create a simple Live Event with one Adobe RTMP output
  - Example XML: Create a more advanced Live Event with 3 streams using presets
  - Example XML: Using a Live Event Profile to create a new Live Event with advanced overrides
- Live Event Profiles
- Schedules
- Presets
- Preset Categories
- MPTS Multiplexers
- Settings
- Alerts and Messages
- Devices
- System Status
- Routers
- Error Codes
- Warning Codes
- Audit Message Codes
- Query Parameters
- Authentication and REST

REST BASICS

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

HTTP HEADERS

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

API VERSIONS

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

SIMPLE EXAMPLES

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

```
curl -H "Accept: application/xml" https://<server_ip>/api/live_events
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<live_event_list>
<live_event href="/live_events/3" \
    version="2.19.2.0.xxxx" product="Elemental Live">
  <input>
    <device_input>
      <channel>1</channel>
      <channel_type>HD-SDI</channel_type>
      <sdi_settings>
        <scte104_offset>0</scte104_offset>
      </sdi_settings>
      <device_type>AJA</device_type>
      <device_number>0</device_number>
    </device_input>
  </input>
  <node_id>2</node_id>
  <notification>
    <email></email>
    <web_callback_url></web_callback_url>
  </notification>
  <user_data></user_data>
  <submitted>2010-01-26 02:07:08 UTC</submitted>
  <status>complete</status>
</live_event>
...
</live_event_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.

For example, to create a basic preset for a stream with H.264 video and AAC audio:

```
curl -H "Accept: application/xml" -H "Content-type: application/xml" \
-d @filename https://<server_ip>/api/v2.19.2.0/presets
```

where the file indicated by filename contains

```
<preset>
  <name>Example Preset</name>
  <video_description>
    <width>1280</width>
    <height>720</height>
    <codec>H.264</codec>
    <h264_settings>
      <bitrate>3000000</bitrate>
    </h264_settings>
  </video_description>
```

```
  <audio_description>
    <codec>AAC</codec>
    <aac_settings>
      <bitrate>128000</bitrate>
    </aac_settings>
  </audio_description>
</preset>
```

returns

```
<?xml version="1.0" encoding="UTF-8"?>
<preset href="/presets/33" \
    version="2.19.2.0.xxxx" product="Elemental Live">
  <name>Example Preset</name>
  <description></description>
  <video_description>
    <anti_alias>false</anti_alias>
    <gpu nil="true"></gpu>
    <height>720</height>
    <id>130</id>
    <width>1280</width>
    <codec>h.264</codec>
    <h264_settings>
      <adaptive_quantization>off</adaptive_quantization>
      <bitrate>3000000</bitrate>
      <buf_fill_pct nil="true"></buf_fill_pct>
      <buf_size nil="true"></buf_size>
      <cabac>false</cabac>
      <framerate_denominator nil="true"></framerate_denominator>
      <framerate_follow_source>true</framerate_follow_source>
      <framerate_numerator nil="true"></framerate_numerator>
      <gop_closed_cadence>1</gop_closed_cadence>
      <gop_num_b_frames>2</gop_num_b_frames>
      <gop_size>80</gop_size>
      <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>
      <par_denominator nil="true"></par_denominator>
      <par_follow_source>true</par_follow_source>
      <par_numerator nil="true"></par_numerator>
      <passes>1</passes>
      <qp nil="true"></qp>
      <scd>true</scd>
      <slices>0</slices>
      <profile>Main</profile>
      <rate_control_mode>CBR</rate_control_mode>
      <interlace_mode>progressive</interlace_mode>
    </h264_settings>
    <video_preprocessors>
    </video_preprocessors>
  </video_description>
  <audio_description>
    <id>126</id>
    <codec>aac</codec>
    <aac_settings>
      <bitrate>128000</bitrate>
      <channels>2</channels>
      <sample_rate>44100</sample_rate>
      <vbr_quality nil="true"></vbr_quality>
      <profile>LC</profile>
      <rate_control_mode>CBR</rate_control_mode>
    </aac_settings>
  </audio_description>
</preset>
```


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 Live Events, or to slightly modify Live Events, Live Event Profiles, or Presets. Therefore, the Elemental Live 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 Live Event 1. Simply make the regular GET request and add an extra parameter clean=true at the end.

```
curl -H "Accept: application/xml" https://<server_ip>/api/live_events/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 Live Event, or the file may be edited to make any necessary adjustments. The clean xml for a Live Event can also be downloaded directly from the web interface -- from the Live Event Control page, click 'Live Event XML'.

SCHEMA DEFINITIONS

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/LiveEvent.xsd - live event creation schema
- /schema/LiveEventProfile.xsd - live event profile creation schema
- /schema/LivePreset.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 "<preset></preset>" https://<server_ip>/api/v2.19.2.0/presets
```

returns

```
<?xml version="1.0" encoding="UTF-8"?>
<errors>
  <error>Video description can't be blank</error>
  <error>Audio description can't be blank</error>
  <error>Name can't be blank</error>
</errors>
```

Errors, warnings, and audit messages for Live Events are indicated by the status, error, warning, and audit message fields returned in the Live Event's status message. Messages include a code and a message. For example:

```
curl -H "Accept: application/xml" https://<server_ip>/api/live_events/58/status
```

on a system where Live Event 58 has some errors returns something like

```
<?xml version="1.0" encoding="UTF-8"?>
<live_event href="https://server_ip:80/live_events/58"
    \version="2.19.2.0.xxxx" product="Elemental Live">
  <user_data></user_data>
  <submitted>2010-02-22 13:46:34 -0700</submitted>
  <status>error</status>
  <pct_complete>16</pct_complete>
```

```
  <average_fps>53.4</average_fps>
  <start_time>2010-02-22 13:46:34 -0700</start_time>
  <complete_time></complete_time>
  <elapsed>835</elapsed>
  <errors>
    <error>
      <code>1999</code>
      <created_at>2010-02-22 14:00:29 -0700</created_at>
      <message>EME timeout detected</message>
    </error>
  </errors>
</live_event>
```

Audit messages provide the user with additional information about the execution of the Live Event.

LIVE EVENTS

The following table describes the REST Live Event control interface.

URL	METHOD	PARAMETERS	RETURNS	DESCRIPTION
/live_events	GET	Pagination parameters, Filter parameters	Live Event List Description	Retrieves a list of Live Events in the system
/live_events	POST	Live Event Parameters	Live Event Description	Creates a new Live Event
/live_events/<id>	GET		Live Event Description	Retrieves a specific Live Event in the system
/live_events/<id>	PUT	Live Event Parameters	Live Event Description	Updates an existing Live Event with new settings. Append with the URL query-param ?unlocked=1 to stop and restart a running event.
/live_events/<id>?unlocked=1&active_input_id=<input id>	PUT	Live Event Parameters	Live Event Details	Stops the event and restarts it with the new settings (specified in the Live Event Parameters) and starting at the input identified by the specified input ID. The input ID must be part of the original event definition; it cannot identify an input that was added using the /live_events/<id>/playlist endpoint.
/live_events/<id>?unlocked=1&active_input_label=<input label>	PUT	Live Event Parameters	Live Event Details	Stops the event and restarts it with the new settings (specified in the Live Event Parameters) and starting at the input identified by the specified input label. The input label must be part of the original event definition; it cannot identify an input that was added using the /live_events/<id>/playlist endpoint.
/live_events/<id>	DELETE			Permanently deletes a Live Event
/live_events/<id>/status	GET		Live Event Status	Retrieves a summary of Live Event <id>'s status, without detailed encoding parameters
/live_events/<id>/priority	GET		Live Event Priority	Retrieves Live Event <id>'s priority
/live_events/<id>/priority	POST	<priority>value</priority>	Live Event Priority	Sets Live Event <id>'s priority
/live_events/<id>/start	POST	<start></start>	Live Event Details	Starts encoding Live Event <id>
/live_events/<id>/start?active_input_id=<input id>	POST	<start></start>	Live Event Details	Starts the event at the specified input ID. The input ID must be part of the original event definition; it cannot identify an input that was added using the /live_events/<id>/playlist endpoint.

URL	METHOD	PARAMETERS	RETURNS	DESCRIPTION
/live_events/<id>/ start?active_input_label=<input label>	POST	<start></start>	Live Event Details	Starts the event at the input identified by specified input label. The input label must be part of the original event definition; it cannot identify an input that was added using the /live_events/<id>/playlist endpoint.
/live_events/<id>/ stop	POST	<stop></stop>	Live Event Details	Stops encoding Live Event <id>
/live_events/<id>/ cancel	POST	<cancel></cancel>	Live Event Details	Cancels pending Live Event <id>
/live_events/<id>/ archive	POST	<archive></archive>	Live Event Description	Marks a Live Event as Archived. Live Event will no longer appear in main /live_events list.
/live_events/<id>/ reset	POST	<reset></reset>	Live Event Details	Resets Live Event <id> back to the Pending state. To avoid accidentally overwriting archive outputs after a reset, use a timestamp d in the destination.
/live_events/<id>/ mute_audio	POST	<mute_audio></mute_audio>		Mutes the audio of Live Event <id>. Sets audio gain to -60.
/live_events/<id>/ unmute_audio	POST	<unmute_audio></unmute_audio>		Unmutes the audio of Live Event <id>. Sets audio gain to 0.
/live_events/<id>/ adjust_audio_gain	POST	<gain>value</gain>		Sets the audio gain for Live Event <id> to value.
/live_events/<id>/ rollover_output	POST	<output_id>output id</output_id>		Output <output id> must be an Archive output that is currently running. Instructs Output <output id> to rollover.
/live_events/<id>/ pause_output	POST	<output_id>output id</output_id>		Output <output id> must be an output that is currently running. Instructs Output <output id> to pause.
/live_events/<id>/ unpause_output	POST	<output_id>output id</output_id>		Output <output id> must be an output that is currently paused. Instructs Output <output id> to resume.
/live_events/<id>/ stop_output	POST	<output_id>output id</output_id>		Output <output id> must be an output that is currently running. Instructs Output <output id> to stop.
/live_events/<id>/ start_output	POST	<output_id>output id</output_id>		Output <output id> must be an output that is currently stopped. Instructs Output <output id> to start.
/live_events/<id>/ pause_output_group	POST	<group_id>output group id</group_id> or <group_name>_name_</group_name>		Output Group <output group id> (or group_name) must be a group that is currently running. Instructs Output Group <output group id> to pause.
/live_events/<id>/ unpause_output_group	POST	<group_id>output group id</group_id> or <group_name>_name_</group_name>		Output Group <output group id> (or group_name) must be a group that is currently paused. Instructs Output Group <output group id> to resume.
/live_events/<id>/ stop_output_group	POST	<group_id>output group id</group_id> or <group_name>_name_</group_name>		Output Group <output group id> (or group_name) must be a group that is currently running. Instructs Output Group <output group id> to stop.
/live_events/<id>/ start_output_group	POST	<group_id>output group id</group_id> or <group_name>_name_</group_name>		Output Group <output group id> (or group_name) must be a group that is currently stopped or paused. Instructs Output Group <output group id> to start.
/live_events/<id>/ stream_metadata	POST	Arbitrary Stream Metadata: any content inside the XML file		Sends the body of the POST request to all Adobe RTMP outputs as metadata (via onUserDataEvent). The proper Content-type header must be used for the body content.

URL	METHOD	PARAMETERS	RETURNS	DESCRIPTION
/live_events/<id>/ amf_metadata	POST	AMF Metadata		Sends the body of the POST request to all Adobe RTMP outputs as AMF metadata via onCuePoint.
/live_events/<id>/ update_timing	POST	Timing Parameters	Timing parameters	Adjusts the timing parameters of a currently running Live Event. This allows the user to contract or extend the end time of a scheduled Live Event.
/live_events/<id>/ cue_point	POST	Cue Point Parameters	Insertion Information	Manually edits an SCTE-35 cue point in the stream. Time given is either system time or preroll. Outputs that have scte35_passthrough enabled will output this cue point. Note that an SDI source using the Embedded timecode source must have embedded timecodes for this feature to work.
/live_events/<id>/ time_signal	POST	Time Signal Parameters	Insertion Information	Manually edits an SCTE-67 time signal in the stream. Time is given in system time. Outputs that have scte35_passthrough enabled will output this time signal. Note that an SDI source using the Embedded timecode source must have embedded timecodes for this feature to work.
/live_events/<id>/ image_inserter	POST	Image Inserter for REST Parameters	Image Inserter REST parameters	Inserts and removes images into image inserter objects that have REST commands enabled
/live_events/<id>/ motion_image_inserter	POST	Motion Image Inserter for REST Parameters	Motion Image Inserter REST parameters	Updates images on the motion image inserter that is REST enabled
/live_events/<id>/ activate_input	POST	<input_id>i_id</input_id> OR <input_label>unique name for input</input_label> OR specify an activation time (in utc) <activate_input> <input_id>i_id</input_id> <utc_time>20151231T235959.999</utc_time> </activate_input>		Switches the currently running Live Event <id> to Input <i_id>, immediately or at the specified time. Format of the UTC time is expressed is YYYYMMDDThhmmss.[frac_second], for example 20151231T235959.999.
/live_events/<id>/ prepare_input	POST	Required: specify the input as <input_id>i_id</input_id> OR <input_label>unique name for input</input_label> Optional: specify the time for the input preparation to begin <utc_time>20151231T23595959.999</utc_time>. The input is prepared immediately when not present. Optional: specify the input should be perpetually prepared or it should only last until the input becomes active. Specify via <perpetual>true</perpetual> or <perpetual>false</perpetual>. The default setting is false. Optional: specify a preparation_index of 0 or 1 (defaults to 0 when not present) <preparation_index>0</preparation_index> If optional fields are provided, the entire payload must be surrounded by <prepare_input> tags, i.e. <prepare_input> <input_id>i_id</input_id> <utc_time>1234556789</utc_time> </prepare_input>.		Prepares the currently running Live Event <id> for switching to Input <i_id> either immediately or at the specified time. Format of the UTC time is expressed is YYYYMMDDThhmmss.[frac_second], for example 20151231T235959.999. The optional preparation_index can be 0 or 1 allowing up to two inputs to be prepared simultaneously. When <perpetual> is set to true the specified input will continue to be prepared until the preparation_index it was prepared with is reused in a subsequent prepare_input command with either another valid input ID, or the special value of 0xffffffff which cancels preparation. When <perpetual> is set to false the specified input will continue to be prepared until the preparation_index it was prepared with is reused in a subsequent prepare_input command with either another valid input ID, or the special value of 0xffffffff which cancels preparation, or the specified input becomes active.
/live_events/<id>/ playlist	POST	<inputs> <input></input>...<input></input> </inputs>		This endpoint causes all non-active inputs to be removed from the running event. The new inputs defined in the payload will be appended after the currently running input.

URL	METHOD	PARAMETERS	RETURNS	DESCRIPTION
/live_events/<id>/ inputs	GET		List of all inputs on the event	Retrieve list of inputs on the event.
/live_events/<id>/ inputs/<input_id>	GET	Input ID	Specified input details	Retrieve full input information for the specified input_id on the specified Live Event.
/live_events/<id>/ inputs	POST	Input tag	Input_id if success	Add an input to a running event. Max inputs is 30. Input tag can be copied from an existing valid Live event XML.
/live_events/<id>/ inputs/<input_id>	PUT	Input tag		Modify an input to a running event. An active input canâ€™t be modified. input_id can be obtained from the Input Controls panel of a running event.
/live_events/<id>/ inputs/<input_id>	DELETE			Delete an input to a running event. An active input canâ€™t be deleted.
/live_events/<id>/ inputs/by_label/<input_label>	PUT	Input tag		Modify an input to a running event. An active input canâ€™t be modified. input_label can be obtained from the Input Controls panel of a running event.
/live_events/<id>/ inputs/by_label/<input_label>	DELETE			Delete an input to a running event. An active input canâ€™t be deleted.
/live_events/<id>/ outputs	GET		List of all outputs on the event	Retrieve list of all outputs on the event.
/live_events/<id>/ reset_video_buffer_stats	POST	<reset_video_buffer_stats> </reset_video_buffer_stats>		Resets the input video buffer summary statistics while a Live Event is running.
/live_events/<id>/ timed_metadata	POST	Timed Metadata Parameters		Inserts timed metadata (ID3 tags) into Apple HLS or MPEG Transport stream outputs with insert_timed_metadata enabled.
/live_events/<id>/ private_metadata	POST	Private Metadata Parameters		Inserts private metadata encoded in AMF into MPEG Transport streams with insert_private_metadata enabled.
/live_events/<id>/ bulk_metadata	POST	<bulk_metadata> Timed Metadata Parameters Private Metadata Parameters </bulk_metadata>		Inserts one or more timed metadata and private metadata items at the same time.
/live_events/<id>/ avail_image	POST	<avail_image> Ad Avail Image Parameters </avail_image>		Replaces the configured ad avail blanking image
/live_events/<id>/ blackout_image	POST	<blackout_image> Blackout Image Parameters </blackout_image>		Replaces the configured blackout image insertion image.
/live_events/<id>/ cut_lists	GET		Live Event Cut Lists	Retrieves a list of Live Event <id>'s cut lists. Cut lists are generated for each output where "Log Edit Points" is true.
/live_events/<id>/ cut_lists/<output_id>	GET		Live Event Output Cut List	Retrieves the cut list XML for Output <output_id> in Live Event <id>. <output_id> can be retrieved by using the above end point to get a list of all cut lists for Live Event <id>.

EXAMPLE XML: CREATE A LIVE EVENT FROM A LIVE EVENT 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 Live Event will not be created.

```
<?xml version="1.0" encoding="UTF-8"?>
<live_event>
  <input>
    <device_input>
      <channel>1</channel>
      <channel_type>HD-SDI</channel_type>
```



```
      <sdi_settings>
        <scte104_offset>0</scte104_offset>
      </sdi_settings>
      <device_id>1</device_id>
    </device_input>
  </input>

  <profile>1</profile>
</live_event>
```

EXAMPLE XML: CREATE A SIMPLE LIVE EVENT WITH ONE ADOBE RTMP OUTPUT

```
<?xml version="1.0" encoding="UTF-8"?>
<live_event>
  <input>
    <device_input>
      <channel>1</channel>
      <channel_type>HD-SDI</channel_type>
      <sdi_settings>
        <scte104_offset>0</scte104_offset>
      </sdi_settings>
      <device_id>1</device_id>
    </device_input>
  </input>

  <stream_assembly>
    <name>stream1</name>
    <video_description>
      <gpu>1</gpu>
      <height>480</height>
      <width>640</width>
      <codec>h.264</codec>
      <h264_settings>
        <adaptive_quantization>medium</adaptive_quantization>
        <bitrate>1000000</bitrate>
        <cabac>true</cabac>
        <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>2</gop_num_b_frames>
        <gop_size>90</gop_size>
        <look_ahead_rate_control>medium</look_ahead_rate_control>
        <scd>true</scd>
        <slices>0</slices>
        <profile>Main</profile>
        <rate_control_mode>CBR</rate_control_mode>
        <interlace_mode>progressive</interlace_mode>
      </h264_settings>
      <video_preprocessors>
        <deinterlacer>
          <deinterlace_mode>Adaptive</deinterlace_mode>
        </deinterlacer>
      </video_preprocessors>
    </video_description>
    <audio_description>
      <codec>aac</codec>
      <aac_settings>
        <bitrate>64000</bitrate>
        <channels>2</channels>
        <sample_rate>44100</sample_rate>
        <vbr_quality>HIGH3</vbr_quality>
        <profile>LC</profile>
        <rate_control_mode>CBR</rate_control_mode>
      </aac_settings>
```

```
    </audio_description>
  </stream_assembly>
  <output_group>
    <type>rtmp_group_settings</type>
    <rtmp_group_settings>
      <cdn>None</cdn>
    </rtmp_group_settings>
    <output>
      <stream_assembly_name>stream1</stream_assembly_name>
      <container>rtmp</container>
      <rtmp_settings>
        <connection_retry_interval>2</connection_retry_interval>
        <stream_name>Stream1</stream_name>
        <rtmp_endpoint>
          <password></password>
          <uri>rtmp://flashmediaserver/live</uri>
          <username></username>
        </rtmp_endpoint>
      </rtmp_settings>
    </output>
  </output_group>
</live_event>
```

Note that stream_assembly sections have a <name> element that is used later in the <output> section to specify an output for a given stream. Once the Live Event is submitted, the system assigns an <id> element to the <stream_assembly> to create a permanent reference, and the <name> element is no longer used.

EXAMPLE XML: CREATE A MORE ADVANCED LIVE EVENT USING PRESETS

```
<?xml version="1.0" encoding="UTF-8"?>
<live_event>
  <input>
    <device_input>
      <channel>1</channel>
      <channel_type>HD-SDI</channel_type>
      <sdi_settings>
        <scte104_offset>0</scte104_offset>
      </sdi_settings>
      <device_id>1</device_id>
    </device_input>
  </input>

  <stream_assembly>
    <name>stream1</name>
    <preset>1</preset>
  </stream_assembly>
  <stream_assembly>
    <name>stream2</name>
    <preset>2</preset>
  </stream_assembly>
  <stream_assembly>
    <name>stream3</name>
    <preset>3</preset>
  </stream_assembly>
  <stream_assembly>
    <name>stream4</name>
    <preset>8</preset>
  </stream_assembly>
  <output_group>
    <type>archive_group_settings</type>
    <archive_group_settings>
      <destination>
        <uri>/data/server/archive</uri>
      </destination>
    </archive_group_settings>
```

```
    <output>
      <extension>mp4</extension>
      <name_modifier>_3mbit</name_modifier>
      <stream_assembly_name>stream4</stream_assembly_name>
      <container>mp4</container>
    </output>
  </output_group>
  <output_group>
    <type>apple_live_group_settings</type>
    <apple_live_group_settings>
      <generate_meta_file>true</generate_meta_file>
      <index_n_segments>10</index_n_segments>
      <keep_segments>20</keep_segments>
      <segment_length>10</segment_length>
      <vod_mode>false</vod_mode>
      <destination>
        <uri>/data/server/apple/live</uri>
      </destination>
    </apple_live_group_settings>
    <output>
      <extension>m3u8</extension>
      <name_modifier>_1450000</name_modifier>
      <stream_assembly_name>stream1</stream_assembly_name>
      <container>m3u8</container>
    </output>
    <output>
      <extension>m3u8</extension>
      <name_modifier>_800000</name_modifier>
      <stream_assembly_name>stream2</stream_assembly_name>
      <container>m3u8</container>
    </output>
    <output>
      <extension>m3u8</extension>
      <name_modifier>_450000</name_modifier>
      <stream_assembly_name>stream3</stream_assembly_name>
      <container>m3u8</container>
    </output>
  </output_group>
</live_event>
```

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

When creating a new Live Event using an existing Live Event Profile, it is sometimes necessary to override specific settings deep within the Live Event Profile to suit the particular needs of your Live Event. 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 Live Event 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 Live Event 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 Live Event. Consider the following example of this workflow for a simple case.

To begin, let's assume you have a Live Event 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"?>
<live_event_profile version="2.19.2.0.xxxx" product="Elemental Live">
  <name>Basic MS Smooth</name>
  <permalink>basic_ms_smooth</permalink>
  <description>One MS Smooth output with DRM</description>
  <loop_all_inputs>false</loop_all_inputs>
  <timecode_config>
    <source>embedded</source>
  </timecode_config>
  <failure_rule>
    <priority>50</priority>
```



```
    <restart_on_failure>false</restart_on_failure>
  </failure_rule>
  <initial_audio_gain>0</initial_audio_gain>
  <input_buffer_size>60</input_buffer_size>
  <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"/>
      <force_cpu_encode>false</force_cpu_encode>
      <height nil="true"/>
      <insert_color_metadata>true</insert_color_metadata>
      <respond_to_afd>None</respond_to_afd>
      <sharpness>50</sharpness>
      <stretch_to_output>false</stretch_to_output>
      <timecode_passthrough>false</timecode_passthrough>
      <vbi_passthrough>false</vbi_passthrough>
      <width nil="true"/>
      <h264_settings>
        <adaptive_quantization>medium</adaptive_quantization>
        <bitrate>5000000</bitrate>
        <buf_fill_pct nil="true"/>
        <buf_size nil="true"/>
        <cabac>true</cabac>
        <flicker_aq>true</flicker_aq>
        <force_field_pictures>false</force_field_pictures>
        <framerate_denominator nil="true"/>
        <framerate_follow_source>true</framerate_follow_source>
        <framerate_numerator nil="true"/>
        <gop_b_reference>false</gop_b_reference>
        <gop_closed_cadence>1</gop_closed_cadence>
        <gop_markers>false</gop_markers>
        <gop_num_b_frames>2</gop_num_b_frames>
        <gop_size>90.0</gop_size>
        <gop_size_units>frames</gop_size_units>
        <interpolate_frc>false</interpolate_frc>
        <look_ahead_rate_control>medium</look_ahead_rate_control>
        <max_bitrate nil="true"/>
        <max_qp nil="true"/>
        <min_bitrate nil="true"/>
        <min_buf_occ nil="true"/>
        <min_i_interval>0</min_i_interval>
        <min_qp nil="true"/>
        <num_ref_frames>1</num_ref_frames>
        <par_denominator nil="true"/>
        <par_follow_source>true</par_follow_source>
        <par_numerator nil="true"/>
        <passes>1</passes>
        <qp nil="true"/>
        <repeat_pps>false</repeat_pps>
        <rp2027_syntax>false</rp2027_syntax>
        <scd>true</scd>
        <sei_timecode>false</sei_timecode>
        <slices>0</slices>
        <slow_pal>false</slow_pal>
        <softness nil="true"/>
        <spatial_aq>true</spatial_aq>
        <svq>0</svq>
        <telecine>None</telecine>
        <temporal_aq>true</temporal_aq>
        <profile>Main</profile>
        <rate_control_mode>CBR</rate_control_mode>
        <interlace_mode>progressive</interlace_mode>
```



```
      </h264_settings>
      <selected_gpu nil="true"/>
      <codec>h.264</codec>
    </video_description>
    <audio_description>
      <language_code nil="true"/>
      <order>1</order>
      <stream_name>audio_1</stream_name>
      <wma2_settings>
        <bitrate>64000</bitrate>
        <channels>2</channels>
        <sample_rate>44100</sample_rate>
      </wma2_settings>
      <codec>wma2</codec>
    </audio_description>
  </stream_assembly>
  <output_group>
    <name nil="true"/>
    <order>1</order>
    <ms_smooth_group_settings>
      <connection_retry_interval>2</connection_retry_interval>
      <content_key>ee939e0d-52ff-4b04-b01a-22e2e51674c6</content_key>
      <custom_attributes/>
      <drm_system>playready</drm_system>
      <encryption_type>AES-128-CTR</encryption_type>
      <event_id/>
      <filecache_duration>300</filecache_duration>
      <follow_segment>false</follow_segment>
      <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_playready_license_server.com</license_url>
      <num_retries>10</num_retries>
      <push_mode>true</push_mode>
      <restart_delay>0</restart_delay>
      <send_eos>true</send_eos>
      <send_stream_manifest>true</send_stream_manifest>
      <timestamp_delta_milliseconds nil="true"/>
      <timestamp_delta_seconds nil="true"/>
      <timestamp_offset>2011-01-01</timestamp_offset>
      <timestamp_offset_today>true</timestamp_offset_today>
      <ui_license_url>http://my_playready_ui_license_server.com</ui_license_url>
      <use_event_id>true</use_event_id>
      <publish_point>
        <uri>http://different_publishing_server/endpoint</uri>
      </publish_point>
    </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>_basic_ms_smooth</name_modifier>
      <order>1</order>
      <start_paused>false</start_paused>
      <container>ismv</container>
      <stream_assembly_name>stream_assembly_0</stream_assembly_name>
    </output>
  </output_group>
</live_event_profile>
```


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

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


```
<?xml version="1.0" encoding="UTF-8"?>
<live_event version="2.19.2.0.xxxx" product="Elemental Live">
  <name>Basic MS Smooth</name>
  <input>
    <network_input>
      <uri>udp://my_video_server:5001</uri>
    </network_input>
  </input>
  <loop_all_inputs>false</loop_all_inputs>
  <timecode_config>
    <source>embedded</source>
  </timecode_config>
  <failure_rule>
    <priority>50</priority>
    <restart_on_failure>false</restart_on_failure>
  </failure_rule>
  <initial_audio_gain>0</initial_audio_gain>
  <input_buffer_size>60</input_buffer_size>
  <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"/>
      <force_cpu_encode>false</force_cpu_encode>
      <height nil="true"/>
      <insert_color_metadata>true</insert_color_metadata>
      <respond_to_afd>None</respond_to_afd>
      <sharpness>50</sharpness>
      <stretch_to_output>false</stretch_to_output>
      <timecode_passthrough>false</timecode_passthrough>
      <vbi_passthrough>false</vbi_passthrough>
      <width nil="true"/>
      <h264_settings>
        <adaptive_quantization>medium</adaptive_quantization>
        <bitrate>5000000</bitrate>
        <buf_fill_pct nil="true"/>
        <buf_size nil="true"/>
        <cabac>true</cabac>
        <flicker_aq>true</flicker_aq>
        <force_field_pictures>false</force_field_pictures>
        <framerate_denominator nil="true"/>
        <framerate_follow_source>true</framerate_follow_source>
        <framerate_numerator nil="true"/>
        <gop_b_reference>false</gop_b_reference>
        <gop_closed_cadence>1</gop_closed_cadence>
        <gop_markers>false</gop_markers>
        <gop_num_b_frames>2</gop_num_b_frames>
        <gop_size>90.0</gop_size>
        <gop_size_units>frames</gop_size_units>
        <interpolate_frc>false</interpolate_frc>
        <look_ahead_rate_control>medium</look_ahead_rate_control>
        <max_bitrate nil="true"/>
        <max_qp nil="true"/>
        <min_bitrate nil="true"/>
        <min_buf_occ nil="true"/>
```



```
        <min_i_interval>0</min_i_interval>
        <min_qp nil="true"/>
        <num_ref_frames>1</num_ref_frames>
        <par_denominator nil="true"/>
        <par_follow_source>true</par_follow_source>
        <par_numerator nil="true"/>
        <passes>1</passes>
        <qp nil="true"/>
        <repeat_pps>false</repeat_pps>
        <rp2027_syntax>false</rp2027_syntax>
        <scd>true</scd>
        <sei_timecode>false</sei_timecode>
        <slices>0</slices>
        <slow_pal>false</slow_pal>
        <softness nil="true"/>
        <spatial_aq>true</spatial_aq>
        <svq>0</svq>
        <telecine>None</telecine>
        <temporal_aq>true</temporal_aq>
        <profile>Main</profile>
        <rate_control_mode>CBR</rate_control_mode>
        <interlace_mode>progressive</interlace_mode>
      </h264_settings>
      <selected_gpu nil="true"/>
      <codec>h.264</codec>
    </video_description>
    <audio_description>
      <language_code nil="true"/>
      <order>1</order>
      <stream_name>audio_1</stream_name>
      <wma2_settings>
        <bitrate>64000</bitrate>
        <channels>2</channels>
        <sample_rate>44100</sample_rate>
      </wma2_settings>
      <codec>wma2</codec>
    </audio_description>
  </stream_assembly>
  <output_group>
    <name nil="true"/>
    <order>1</order>
    <ms_smooth_group_settings>
      <connection_retry_interval>2</connection_retry_interval>
      <content_key>ee939e0d-52ff-4b04-b01a-22e2e51674c7</content_key>
      <custom_attributes/>
      <drm_system>playready</drm_system>
      <encryption_type>AES-128-CTR</encryption_type>
      <event_id/>
      <filecache_duration>300</filecache_duration>
      <follow_segment>false</follow_segment>
      <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_playready_license_server.com</license_url>
      <num_retries>10</num_retries>
      <push_mode>true</push_mode>
      <restart_delay>0</restart_delay>
      <send_eos>true</send_eos>
      <send_stream_manifest>true</send_stream_manifest>
      <timestamp_delta_milliseconds nil="true"/>
      <timestamp_delta_seconds nil="true"/>
      <timestamp_offset>2011-01-01</timestamp_offset>
```



```
      <timestamp_offset_today>true</timestamp_offset_today>
      <ui_license_url>http://my_playready_ui_license_server.com</ui_license_url>
      <use_event_id>true</use_event_id>
      <publish_point>
        <uri>http://different_publishing_server/endpoint</uri>
      </publish_point>
    </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>_basic_ms_smooth</name_modifier>
      <order>1</order>
      <start_paused>false</start_paused>
      <container>ismv</container>
      <stream_assembly_name>stream_assembly_0</stream_assembly_name>
    </output>
  </output_group>
</live_event>
```

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

LIVE EVENT PROFILES

Live Event Profiles can be used for commonly used Live Event settings. The permalink of a Live Event Profile may be substituted for its id.

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

SCHEDULES

Schedules can be created to run certain Live Event Profiles at scheduled times, or a set of repeating times.

URL	METHOD	PARAMETERS	RETURNS	DESCRIPTION
/schedules	GET	Pagination parameters	Schedule List	Retrieves a list of Schedules in the system
/schedules	POST	Schedule Parameters	Schedule Description	Creates a new Schedule
/schedules/<id>	GET		Schedule Description	Retrieves a specific Schedule in the system
/schedules/<id>	PUT	Schedule Parameters	Schedule Description	Updates an existing schedule with new settings
/schedules/<id>	DELETE			Deletes schedule <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>

MPTS MULTIPLEXERS

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

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

URL	METHOD	PARAMETERS	RETURNS	DESCRIPTION
/mpts/<mpts_id>/mpts_members	POST	MPTS Member Parameters	MPTS Member XML of newly-created channel, or errors	Add a new channel member to an MPTS
/mpts/<mpts_id>/mpts_members/<mpts_member_id>	PUT	MPTS Member Parameters	MPTS Member XML of updated channel, or errors	Update an existing channel member of an MPTS
/mpts/<mpts_id>/mpts_members/<mpts_member_id>	DELETE		Successful response, or errors	Remove an existing channel member of an MPTS

EXAMPLE XML: CREATE AN MPTS

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

SETTINGS

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

URL	METHOD	RETURNS	DESCRIPTION
/settings	GET	Timezone, Network Settings, Firewall Settings, Mount Point Settings, Authentication Settings, Sequencer Settings, (Cluster Settings if part of a cluster)	Retrieves information about the current system settings. This XML is in a format that is accepted by the configure script (-i <filename>). This can be used to configure many identical boxes.
/settings?cluster=true	GET	Timezone, Network Settings, Firewall Settings, Mount Point Settings, Authentication Settings, Sequencer Settings, Cluster Settings	Retrieves information about the current system settings including example cluster settings configured such that the current server is the master node. This can be used to help configure Slave nodes after the Master node has been configured.
/settings/network	GET	Network Settings	Retrieves information about the current network settings. Other Elemental Server units can communicate on the interface marked <management_interface> in a clustered environment.
/settings/mount_points	GET	Mount Point Settings	Retrieves information about the current mount point settings.
/settings/firewall	GET	Firewall Settings	Retrieves information about the current firewall settings.
/settings/snmp	GET	SNMP Settings	Retrieves information about the current SNMP settings.
/settings/authentication	GET	Authentication Settings	Retrieves information about the current authentication settings.
/settings/advanced	GET	Sequencer Parameters	Retrieves information about the current sequencer settings. Only available for clustered systems.

URL	METHOD	RETURNS	DESCRIPTION
/settings/stop	POST	<stop></stop>	Enables a graceful shutdown of the Elemental service. Currently running Live Events will run to completion, but no new Live Events will be started. When all Live Events have completed the service will shut down.
/settings/start	POST	<start></start>	Sends a start command to the Elemental service. Used to restart the service after a /settings/stop command.
/settings/advanced	POST	Sequencer Parameters	Updates Sequencer settings. Requires a restart of the Service in order to take effect.

ALERTS AND MESSAGES

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

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

DEVICES

Devices are auto-detected on the node, so there is no POST. You can PUT to add an optional <name>. GET returns several read-only attributes, including information about the router and router input that the device is connected to, if it is in fact connected to a device. To connect the device to a router, see the Router and Router Output entities.

GET and GET List return slightly different information; if you cannot find the information you are looking for with one command, then try the other.

URL	METHOD	PARAMETERS	RETURNS	DESCRIPTION
/devices	GET		List of input devices (HD-SDI inputs)	Retrieves available input devices in the system.
/devices/<id>	GET		Device Description	Retrieves a device from the system.
/devices/<id>	PUT	Device Parameters	Device Description	Updates an existing device with the new settings.
/devices/<id>/preview	POST	Preview image list (see example below)	Preview image locations	Generates jpeg images of the specified sizes from the current input on device <id>.

RESPONSE INFORMATION

NAME	TYPE	DESCRIPTION
device_number	Integer	Read-only. The port number, as specified by the card itself.
device_type	String	Read-only. The manufacturer's type for the card, as specified by the card itself.
id	Integer	Read-only. The ID for this port on this ID card, unique among all cards on this node: it is the one assigned by the system and is the ID that the system uses to identify the port.
name	String	Read-write. A name you assign to the port. May be null.
quad	String	Read-only. False: This port is a physical port on the card. True: It is a virtual port to handle quad-link input, and will only appear if the card supports quad-link input.

NAME	TYPE	DESCRIPTION
node_id	String	Read-only. If this node is part of a cluster, the ID for the node in that cluster. If this node is not part of a cluster, always 1.
<router>		All are read-only. Same info as from GET Router. All read-only. These attributes are present only if the device is connected to an output on a router. The <router><id> identifies the ID (as assigned by Elemental) of the connected router.
router <id>	String	The ID for this router, assigned by the system when the router is created.
Other attributes		See GET Router.
router_output		All are read-only. Same info as from GET Router Output. These attributes are present only if the device is connected to an output on a router. The <router_output><id> identifies the ID (as assigned by Elemental) of the connected output on the router specified by <router>.
router_output <id>		An ID for this router output that is unique on this router.
router_output <router_id>	Integer	An ID for the router that this card is connected to. Identical to <router><id>
router_output <output_number	Integer	The ID of the router output that is connected to the SDI card identified by device_id (below). This ID is the ID assigned by the router (not as assigned by Elemental). Identical to <router><output_number>
router_output <device_id	Integer	Identical to <device><id>
router_output <device_type>	String	Identical to <device><device_type>

EXAMPLE XML: GET PREVIEW IMAGES FROM DEVICE 1

```
<?xml version="1.0" encoding="UTF-8"?>
<preview_images>
  <preview_image>
    <width>320</width>
    <height>240</height>
    <name>small.jpg</name>
  </preview_image>
  <preview_image>
    <width>720</width>
    <height>480</height>
    <name>large.jpg</name>
  </preview_image>
</preview_images>
```

returns

```
<preview_images>
  <preview_image>
    <name>small.jpg</name>
    <width>320</width>
    <height>240</height>
    <location>/images/preview/small.jpg</location>
  </preview_image>
  <preview_image>
    <name>large.jpg</name>
    <width>720</width>
    <height>480</height>
    <location>/images/preview/large.jpg</location>
  </preview_image>
</preview_images>
```

The preview images will be available at **https://<server_ip>/images/preview/<name.jpg>**

SYSTEM STATUS

The system status API allows the user to query status of input devices when running, RAID status when available, and PSU status when available. In addition, basic system information such as memory, CPU and GPU load are displayed when the service is running.

URL	METHOD	PARAMETERS	RETURNS	DESCRIPTION
/system_status.json	GET		System information in JSON format	Retrieves available system statuses.
/system_status.xml	GET		System information in XML format	Retrieves available system statuses.

EXAMPLE JSON: GET SYSTEM STATUS


```
{
  "cpu":
    {"pct":"1.46","num":0},
  "mem":
    {"pct":48,"num":5},
  "gpu":
    [{"pct":0,"num":0}],
  "status":"orange_status",
  "date":"Jan 20, 2015 16:01:47",
  "devices":[],
  "psu_status":"Status: Present, OK",
  "raid_status":"There was an error while trying to get information about the device."
}
```


ROUTERS

The following table describes the REST Router control interface.

URL	METHOD	RETURNS	DESCRIPTION
/routers	POST	POST Router	Create a new router.
/routers/<router_id>	PUT	PUT Router	Modify the attributes of the specified router.
/routers	GET	GET Router List	Get the list of routers.
/routers/<router_id>	GET	GET Router	Get the attributes of the specified router.
/routers/<router_id>	DELETE	DELETE Router	Delete the specified router.
/routers/<router_id>/inputs	POST	POST Router Input	Create a new input for the specified router.
/routers/<router_id>/inputs/input-<input_number>	PUT	PUT Router Input	Modify the attributes of the specified input on the specified router.
routers/<router_id>/inputs	GET	GET Router Input List	Get the list of inputs for the specified router.
/routers/<router_id>/inputs/input-<input_number>	GET	GET Router Input	Get the attributes of the specified input on the specified router.
/routers/<router_id>/inputs/input-<input_number>	DELETE	DELETE Router Input	Delete the specified input on the specified router
/routers/<router_id>/outputs	POST	POST Router Output	Create a new output for the specified router.
/routers/<router_id>/outputs/output-<output_number>	PUT	PUT Router Output	Modify the attributes of the specified output on the specified router.
/routers/<router_id>/outputs	GET	GET Router Output List	Get the list of outputs for the specified router.
/routers/<router_id>/outputs/output-<output_number>	GET	GET Router Output	Get the attributes of the specified output on the specified router.
/routers/<router_id>/outputs/output-<output_number>	DELETE	DELETE Router Output	Delete the specified output on the specified router

WORKING WITH ROUTERS

POST ROUTERS

HTTP URL
POST /routers

BODY OF HTTP

XML content consisting of one router element, consisting of the following tags:

TAG	VALUE	DESCRIPTION
name	String	A name of your choosing.
ip	String	The IP address without any protocol.
max_inputs	Integer	Typically, the number of physical inputs on the router.
max_outputs	Integer	Typically, the number of physical outputs on the router.
router_type	String	One of: • blackmagic_videohub • miranda_nvision • harris_panacea • snell_aurora
level_id	Integer	Appears only for Harris Panacea, Miranda nVision and Snell Aurora.
user_id	Integer	Appears only for Miranda nVision and Snell Aurora.
matrix_id	Integer	

EXAMPLE
Request

This request creates one router with the name "BlackMagic1". This router has 12 inputs and 12 outputs, so these numbers are set in the max_inputs and max_outputs.

```
POST https://<server_ip>/routers
--------------------------------------------------------------
<router>
    <name>BlackMagic1</name>
    <ip>192.168.10.10</ip>
    <router_type>blackmagic_videohub</router_type>
    <max_inputs>12</max_inputs>
    <max_outputs>12</max_outputs>
</router>
```

Response

In this example, the router is given the ID 1.

```
<?xml version="1.0" encoding="UTF-8"?>
<router href="#routers" product="Live" version="n.n.n.123456">
  <id>1</id>
  <name>BlackMagic1</name>
  <ip>192.168.10.10</ip>
  <router_type>blackmagic_videohub</router_type>
  <max_inputs>12</max_inputs>
  <max_outputs>12</max_outputs>
<router>
```

PUT ROUTER

Modify the attributes of the specified router. You cannot use this command to modify the inputs or outputs of the router; to do that, use PUT Router Input and PUT Router Output.

HTTP URL
/routers/<ID of router>

BODY OF HTTP

The body contains only the tags to change; see POST.

GET ROUTER LIST

Get a list of all video SDI routers, including the data that is contained in the Router Input and Router Output entities.

HTTP URL
GET /routers

RESPONSE

XML content consisting of one routers element that contains:

- A HREF attribute that specifies the product and version installed on the node.
- Zero or more router elements, one for each router found. Each element consists of several tags:

TAG	VALUE	DESCRIPTION
id	Integer	The ID for this router, assigned by the system when the router is created.
name	String	See GET Router List.
ip	String	
max_inputs	Integer	
max_outputs	Integer	
router_type	String	
level_id	Integer	
user_id	Integer	
matrix_id	Integer	
inputs		Contains one or more input elements, one for each router input in the router. See GET Router Input List.
outputs		Contains one or more output elements, one for each router output in the router. See GET Router Output List.

EXAMPLE

```
GET https://<server_ip>/routers
-----------------------------------
<?xml version="1.0" encoding="UTF-8"?>
<routers href="#routers">
 <router>
    <id>1</id>
    <name>BlackMagic1</name>
    <ip>192.168.10.10</ip>
    <router_type>blackmagic_videohub</router_type>
    <max_inputs>12</max_inputs>
    <max_outputs>12</max_outputs>
    <inputs>
      <input>
        <id>1</id>
        <name>Input 1</name>
        <router_id>1</router_id>
        <input_number>1</input_number>
      </input>
```



```
          <input>
            <id>2</id>
            <name>Input 2</name>
            <router_id>1</router_id>
            <input_number>2</input_number>
          </input>
        </inputs>
        <outputs>
         <output>
           <id>9</id>
           <router_id>1</router_id>
           <output_number>1</output_number>
           <device_id>1</device_id>
           <device_type>Device</device_type>
         </output>
        </outputs>
      </router>
    .
    .
    .
      <router>
    .
    .
    .
     </router>
    </routers>
```

GET ROUTER

Get the attributes of the specified video SDI router, including the data that is contained in the Router Input and Router Output

HTTP URL

GET /routers/<ID of router>

RESPONSE

XML content consisting of one router element, containing the same tags as the response for GET Router List, above.

DELETE ROUTER

Deletes the specified router (identified by its internal router ID) and the associated inputs and outputs. To get the internal router ID of a specific router, use GET Router.

HTTP URL

DELETE /routers/<ID of router>

WORKING WITH ROUTER INPUTS

The Router Inputs entity holds information about the router inputs that are in use.

POST ROUTER INPUT

Create a new input for the specified router. Use this command repeatedly to set up all the router inputs.

HTTP URL

```
POST   /routers/<ID of router>/inputs
```

BODY OF HTTP

XML content consisting of one input element, consisting of the following tags:

TAG	VALUE	DESCRIPTION
name	String	A name for this input, of your choosing. This name will appear in the Inputs dropdown list on the Elemental Live web interface.
input_number	Integer	The ID of the router input that you want to make known to Elemental Live. This is the ID for the input as assigned by the router (not as assigned by Elemental Live).

RESPONSE

The response repeats back the data that you posted with the addition of:

- id: The newly assigned ID for the router.
- router_id: The router that this router input belongs to.

The response is identical to the response to a GET Router Input. See below for an example.

EXAMPLE

Request

This request creates one input for the router with the ID 2. This input has been automatically assigned the ID 5.

```
POST https://<server_ip>/routers/2/inputs
---------------------------------
<input>
  <name>Input 1</name>
  <input_number>1</input_number>
</input>
```

Response

In this example, the router input has been automatically assigned the ID 4.

```
<?xml version="1.0" encoding="UTF-8"?>
<input href="#inputs" product="Live" version="n.n.n.123456">
  <id>4</id>
  <name>Input 1</name>
  <router_id>2</router_id>
  <input_number>1</input_number>
</input>
```

PUT ROUTER INPUT

Modify the attributes of the specified input on the specified router. If, after the initial setup, you ever change the cabling on the input side of your router, you must use PUT Router Input to reflect these changes.

HTTP URL

```
PUT    /routers/<ID of router>/inputs/<ID of input>
```

BODY OF HTTP

The body contains only the tags to change; see POST, above.

EXAMPLE

This request changes the input with the ID 3. This input belongs to the router with the ID 2. It changes its input_ID to 4; this change would only be made to fix an error in the original setup or to reflect a change in the cabling (so that the router's 4th input is now being used).

```
PUT https://<server_ip>/routers/2/inputs/3
-----------------------------------------------------------
<?xml version="1.0" encoding="UTF-8"?>
```

```
<input>
  <input_id>4</input_id>`
</input>
```

GET ROUTER INPUT LIST

Get the list of all the inputs for the specified router.

HTTP URL

GET /routers/<ID of router>/inputs

RESPONSE

XML content consisting of one inputs element that contains:

- An HREF attribute that specifies the product and version installed on the node.
- Zero or more input elements, one for each input found. Each element consists of several tags:

TAG	VALUE	Description
id	Integer	The ID for this input, unique for this router and assigned by the system when the input is created.
name	String	The name for this input.
router_id	Integer	The router that this input belongs to
input_number	Integer	The ID of the router input as assigned by the router (not as assigned by Elemental Live).

EXAMPLE

This request is for the inputs associated with the router with the ID 4. The response specifies that there are two inputs in this router with IDs 3 and 4.

```
GET https://<server_ip>/routers/4/inputs
----------------------------------------
<?xml version="1.0" encoding="UTF-8"?>
<inputs href="#inputs" product="Live" version="n.n.n.123456">
 <input>
    <id>3</id>
    <name>Input 1</name>
    <router_id>2</router_id>
    <input_number>1</input_number>
 </input>
 <input>
    <id>4</id>
    <name>Input 2</name>
    <router_id>2</router_id>
    <input_number>2</input_number>
 </input>
</inputs>
```

GET ROUTER INPUT

Get the attributes of the specified input on the specified router.

HTTP URL

GET /routers/<ID of router>/inputs/<ID of input>

RESPONSE

XML content consisting of one input element, containing the same tags as the response for GET Router Input List, above.

EXAMPLE

This request get the attributes for the input that has the ID 3. The input belongs to the router with the ID 4.

```
GET https://<server_ip>/routers/4/inputs/3
------------------------------------------------------------
```

```
<?xml version="1.0" encoding="UTF-8"?>
<input href="#3" product="Live" version="n.n.n.123456">
  <id>3</id>
  <name>Input 1</name>
  <router_id>4</router_id>
  <input_number>1</input_number>
</input>
```

DELETE ROUTER INPUT

Delete the specified input on the specified router. To get the ID of the input, use GET Router List and look for the <id> element (not the <input_number> element!).

HTTP URL

```
DELETE /routers/<ID of router>/inputs/<ID of input>
```

WORKING WITH ROUTER OUTPUTS

The Router Outputs entity holds information about each router output that is connected to an SDI card on an Elemental Live hardware unit. The information is a mapping from the router output to the SDI input.

POST ROUTER OUTPUT

Create a new output for the specified router. Use this command repeatedly to set up all the router outputs. The total number of Router Output entities for a specified router must not exceed the max_input on that Router entity: it must not exceed the actual physical outputs on the router.

HTTP URL

```
POST   /routers/<ID of router>/outputs
```

BODY OF HTTP

XML content consisting of one output element, consisting of the following tags:

TAG	VALUE	DESCRIPTION
output_number	Integer	The ID of the router output that is connected to an SDI card on the Elemental Live node. Specify the router by its output ID as assigned by the router (not as assigned by Elemental Live).
device_id	String	The ID the SDI input that is connected to the router output identified by output_number. This ID is assigned by Elemental Live when the SDI card is auto-detected. Each port on the entire node is assigned a unique ID (for example, IDs 1-4 on the first card, 5-8 on the second. Use GET Devices on the Elemental Live API on the individual Elemental Live node to get a list of IDs. (Note that this Elemental Live API does not expose this Device entity.)

RESPONSE

The response repeats back the data that you posted with the addition of:

- id: The newly assigned ID for the router.
- router_id: The router that this router output belongs to.

The response is identical to the response to a GET Router Output. See below for an example.

EXAMPLE

Request

This request creates one output for the router with the ID 2. It specifies that router output ID 3 is connected to SDI card input ID 4.

```
POST https://<server_ip>/routers/2/outputs
----------------------------------
<output>
```



```
  <output_number>3</output_number>
  <device_id>4</device_id>
</output>
```

Response

In this example, the router output is given the ID 4.

```
<?xml version="1.0" encoding="UTF-8"?>
<output href="#output" product="Live" version="n.n.n.123456">
  <id>4</id>
  <output_number>1</output_number>
  <router_id>2</router_id>
  <device_id>1</device_id>
</output>
```

PUT ROUTER OUTPUT

Modify the attributes of the specified output on the specified router. Modify the attributes if a cable from the router output moves to a different SDI card input.

HTTP URL

PUT /routers/<ID of router>/outputs/<ID of output>

BODY OF HTTP

The body contains only the tags to change; see POST, above.

EXAMPLE

This request changes the router output with the ID 4. This input belongs to the router with the ID 2. It changes its device_id to 3, to indicate that the connection represented by this router output is actually to the SDI input that has the ID 3.

```
PUT https://<server_ip>/routers/2/outputs/4
-----------------------------------------------------------
<?xml version="1.0" encoding="UTF-8"?>
<output>
  <device_ID>3</device_id>
</output>
```

GET ROUTER OUTPUT LIST

Get the list of outputs for the specified router.

HTTP URL

```
GET    /routers/<ID of router>/outputs
```

RESPONSE

XML content consisting of one outputs element that contains:

- An HREF attribute that specifies the product and version installed on the node.
- Zero or more output elements, one for each input found. Each element consists of several tags:

TAG	VALUE	DESCRIPTION
id	Integer	The ID for this output, unique for this router and assigned by the system when the output is created.
output_number	Integer	The ID of the router output that is connected to the SDI card identified by device_id (below). This ID is the ID assigned by the router (not as assigned by Elemental Live).
router_id	Integer	The router that this output belongs to, a unique ID assigned by Elemental Live.
device_id	String	The ID the SDI input that is connected to the router output identified by output_number. This ID is assigned by Elemental Live when the SDI card is auto-detected. Each port on the entire node is assigned a unique ID (for example, IDs 1-4 on the first card, 5-8 on the second.

TAG	VALUE	DESCRIPTION
device_type		Always specifies â€œDeviceâ€?

EXAMPLE

This request is for the outputs associated with the router with the ID 2. The response specifies that there are two outputs in this router with IDs 3 and 4.

```
GET https://<server_ip>/routers/2/outputs
---------------------------------------
<?xml version="1.0" encoding="UTF-8"?>
<outputs href="#outputs" product="Live"version="n.n.n.123456">
  <output>
    <id>3</id>
    <output_number>5</output_number>
    <router_id>2</router_id>
    <device_id>1</device_id>
  </output>
  <output>
    <id>4</id>
    <output_number>2</output_number>
    <router_id>2</router_id>
    <device_id>7</device_id>
  </output>
</outputs>
```

GET ROUTER OUTPUT

Get the attributes of the specified output on the specified router.

HTTP URL

```
GET    /routers/<ID of router>/outputs/<ID of output>
```

RESPONSE

XML content consisting of one output element, containing the same tags as the response for GET Router Output List, above.

EXAMPLE

This request get the attributes for the output that has the ID 3. The input belongs to the router with this ID 4.

```
GET https://<server_ip>/routers/4/outputs/3
----------------------------------------------------------
<?xml version="1.0" encoding="UTF-8"?>
<output href="#3" product="Live"version="n.n.n.123456">
  <id>3</id>
  <router_id>4</router_id>
  <output_number>1</output_number>
  <device_id>1</device_id>
  <device_type>Device</device_type>
</output>
```

DELETE ROUTER OUTPUT

Delete the specified output on the specified router.

HTTP URL

```
DELETE https://<server_ip>/routers/2/outputs/1
```

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 Elemental Media Engine (EME) was unable to process an input.
1020	No video stream in input file.	Elemental products require at least one video stream in the input file. Audio only streams are not supported.
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 Live Event. Details are given in the error message. If you see this error, please contact an Elemental support technician with details.
1050	Apple HTTP Push > *n* errors	Pushing Apple streams over HTTP had more than num_retries errors. See live_runner.output for HTTP error codes
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.
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.
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 Elemental process.
1999	*Critical EME (Elemental Media Engine) error*	This code is returned for errors that require an Elemental support technician to continue troubleshooting.
2010	Live Event *live event.id* has too many outputs to run successfully. Please split the outputs into two or more Live Events.	The Live Event will require too many resources to run as one instance. It should be broken up into smaller Live Events so the system can work on the Live Event 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 Live Event.
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.
2060	Live Event *id* was running past its end time	A Live Event was running when the server recovered from an un-clean shutdown (possible power outage). The Live Event was not restarted because it was past its scheduled end time.
2070	Device *name* is already in use	A Live Event attempted to run while its assigned input device was in use by another Live Event.
2072	No available Device	A Live Event with a Router input could not find an available Device

CODE	ERROR MESSAGE	TROUBLESHOOTING
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 Elemental support technician to continue troubleshooting.

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 Live Event, 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 Live Event, 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 Live Event returned a message.
102040	This Live Event is being updated with timestamps in the future relative to the management node.	The node where the Live Event is running has a different system time than the management node. This can cause errors with managing stale Live Events. 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 Live Event, then the system logs an error
102060	Live Event *id* was in running state, but its process was not found (possible power failure). Restarting Live Event	Elemental Live will start any Live Event it finds was running after an un-clean shutdown (such as a power failure). This warning will be logged in that case.
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.

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.
42	Encode Profile and Level	The encoder will report actual profile and level settings for each stream assembly.

QUERY PARAMETERS

The Elemental Live 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:

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

The number of objects per page is capped at 100, any request for a greater number will only return the first 100. To access more than the capped number, use the per_page and page parameter in conjunction to walk the list with multiple requests.

```
/live_events?page=1&per_page=100
/live_events?page=2&per_page=100
/live_events?page=3&per_page=100
/live_events?page=4&per_page=100
```

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" "https://<conductor_id>/jobs?page=3&per_page=30"
```

FILTER

Live Events 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:

```
/live_events?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	Live Events in the pending state
active	Live Events in the preprocessing, running or postprocessing state
pre	Live Events in the preprocessing state
running	Live Events in the running state
post	Live Events in the postprocessing state
complete	Live Events in the complete state
cancelled	Live Events in the cancelled state
error	Live Events in the error state
archived	Live Events that have been archived

AUTHENTICATION AND REST

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

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

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

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

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

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

For example, consider a GET request to https://<server_ip>/api/live_events/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 '/live_events/1' and the X-Auth-Expires value is '1306886400'. Thus the value of X-Auth-Key should be computed as follows:


```
md5('1acpJN7oEDn3BDDYhQ' + md5('/live_events/1'+'admin'+'1acpJN7oEDn3BDDYhQ'+'1306886400'))
=> md5('1acpJN7oEDn3BDDYhQ' + md5('/live_events/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 '/live_events/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/live_events/1
```

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

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

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


LIVE EVENT PARAMETERS

PARAMETERS

The following tables outline parameters that can be set for objects in Elemental Live. 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**.

LIVE EVENT

NAME	TYPE	RANGE	DESCRIPTION
name	string		A name for the Live Event. If left blank the name of the first input will be used.
node	string	Valid node ID or hostname or IP Address	Node on which to run the Live Event when created on a Conductor Live cluster.
input	Input		Live Event input parameters. There can be multiple inputs in a single Live Event.
timecode_config	Timecode Config		Contains settings used to acquire and adjust timecode information from inputs.
loop_all_inputs	boolean	true or **false**	Process list of inputs sequentially and loop from the first input when complete.
failure_rule	Failure Rule		Live Event failover parameters.
profile	string	Valid Profile ID, name, or permalink	This parameter is used only to create an event from the REST API. Include this parameter in order to build the event using a profile. Specify the ID or name or permalink of a profile. When this profile parameter is included and the referenced profile includes an <input> parameter, then the entire event must contain only this profile parameter. When this profile parameter is included and the referenced profile does not include an <input> parameter, the event contains a <profile> and an <input>. When this profile parameter is not included, the event must contain every (relevant) parameter tag.
timing	Timing		Settings for start and end times.
ad_trigger	string	**scte35_splice_insert**, scte35_time_signal_apos, esam	Controls which types of SCTE signals signal Ad Avails. Ads can be signaled with “Splice Insert” messages, which is traditional, or with “Time Signal” messages, carrying “Ad Placement Opportunity Start” segmentation messages (type_id 0×35). See SCTE 35 2013 for more information. In ESAM mode, a POIS server is used for conditioning all SCTE-35 messages and inserting conditioned signals.
ignore_web_delivery_allowed_flag	boolean	true or **false**	When enabled, Segment Descriptors with web_delivery_allowed_flag set to 0 will no longer trigger blackouts or Ad Avail slates
ignore_no_regional_blackout_flag	boolean	true or **false**	When enabled, Segment Descriptors with no_regional_blackout_flag set to 0 will no longer trigger blackouts or Ad Avail slates
initial_audio_gain	integer	-60 to 60 dB (Default: **0 dB**)	Value to set the initial audio gain for the Live Event. This is also editable while the Live Event is running.
avsync_enable	boolean	**true** or false	Enables A/V sync.
avsync_pad_trim_audio	boolean	**true** or false	Enables A/V sync trim audio.
input_loss_behavior	Input Loss Behavior		Settings for system actions when input is lost.
input_end_action	string	switch_input, or none	Indicates the action to take when an input completes (e.g. end-of-file.) Options include immediately switching to the next sequential input (via “switch_input”) or transcoding black / color / slate images per the “Input Loss Behavior” configuration until an activate_input REST command is received (via “none”).

NAME	TYPE	RANGE	DESCRIPTION
output_timing_source	string	input_clock, or system_clock	Indicates whether the rate of frames emitted by the Live encoder should be paced by its system clock (which optionally may be locked to another source via NTP) or should be locked to the clock of the source that is providing the input stream.
input_buffer_size	integer	4 – 300 (Default: **60**)	Number of frames to buffer at input. Higher values will allow less dropped frames, but use more memory. Lower values can improve streaming latency.
resource_reservation	string	none or 4k_decode	When 4K Decode is selected, the system reserves additional resources to provide real-time 4K decode of a second network input. This option allows the system to reserve resources for 4K seamless input switching or 4K Hot-Hot redundancy when network (IP) sources are used. Resource reservation is not required when using Quad SDI sources for 4K encoding. Note – this option only has effect for 4K workflows hosted on the AWS Elemental Live L700AE series (or greater) when 4K encoding is configured. The REST parameter controlled by this checkbox is resource_reservation. It supports values of none and 4k_decode.
low_framerate_input	boolean	true or **false**	Adjusts video input buffer for streams with very low video framerates. This is commonly used for music channels with less than one video frame per second.
low_latency_mode	boolean	true or **false**	Reduces latency of audio/video sync. This reduces overall latency of Live Event, but may result in more dropped audio packets on input timestamp discontinuities. Parameter values such as B Frame and Min-I Interval may still increase latency while Low Latency is set, but the net effect is an overall reduction.
notification	Notification		Settings for notification on status changes of this Live Event.
user_data	string		User-defined data to be attached to the Live Event. This data is available with Live Event status requests via the API.
extract_sdt	boolean	true or **false**	Extracts SDT information from input stream. Displays Service Provider and Service Names during running state.
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 the Live Event, inserts into the decoded input and appears in every output.
motion_image_inserter	Motion Image Inserter		Settings for the motion image inserter. When attached to the Live Event, inserts into the decoded input and appears in every output.
avail_blanking	Avail Blanking		Settings for ad avail blanking.
blackout_slate	Blackout Slate		Settings for blackout slate.
esam	ESAM		Settings for Event Signaling And Messaging (ESAM).
nielsen_configuration	Nielsen Configuration		Nielsen configuration settings
output_lock	OutputLock		Settings for output locking.
output_group	Output Group		Output groups for this Live Event. Output groups contain information about where streams should be distributed.
stream_assembly	Stream Assembly		Stream assemblies for this Live Event. A Live Event can have several stream assemblies which define output codec settings.
ad_avail_offset	integer	-1000 – 1000 (Default: **0**)	When specified, this offset (in milliseconds) is added to the input Ad Avail PTS time. This only applies to embedded SCTE 104/35 messages and does not apply to OOB messages.

LOCATION

```
<live_event>
  ...
  <output_group>
    <type>archive_group_settings</type>
    <archive_group_settings>
      <destination>
        <uri>/data/server/archive/outfile</uri>
      </destination>
    </archive_group_settings>
    ...
  </output_group>
</live_event>
```

NAME	TYPE	RANGE	DESCRIPTION
uri	string		Uniform Resource Identifier – This should be a path to a file accessible to the Elemental Live system, either on the local filesystem or through a SMB mount, or a URI depending on the output type. For example, a rtmp_endpoint should have a uri simliar to: “rtmp://fmsserver/live”.
username	string		Username if credentials are required to access file or publishing point.
password	string		Password if credentials are required to access file or publishing point.
check_server_certificate	boolean	**true** or false	Check HTTPS server certificates. When disabled, we will still check the cryptography in the certificate, but we will not validate the server’s name. Certain subdomains (notably S3 buckets that use dots in the bucket name) do not strictly match the corresponding certificate’s wildcard pattern and would otherwise cause the event to error. This setting is ignored for other protocols.

URI TYPES

URIs identify the location of input sources (both streaming and file sources), output destinations, and assets such as graphics files and scripts.

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
UDP	udp://<hostname>[:port]	None
RTP	rtp://<hostname>[:port]	None
Zixi	Zixi://<hostname>[:port]	None
AWS Elemental MediaConnect	arn://<arn>	Enter the Access Key ID in the username field. Enter the Secret Access Key in the password field.
RTMP	rtmp://<hostname>[:port]	None
Local file	/data/server/folder/file.ext	None
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).

PROTOCOL	URI FORMAT	AUTHENTICATION
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
AWS Elemental MediaStore	https://<container>/<folder>	Enter the Access Key ID in the username field. Enter the Secret Access Key in the password field. Example: https://fje9x3myzr37b9.data.mediastore.us-west-2.amazonaws.com/live_event
Aspera	aspera://<url>	Enter a username and password as specified by the server administrator.

PROTOCOLS FOR NETWORK (STREAMING) INPUTS

Inputs that are network (streaming) inputs can use the HTTP/HTTPS (only for HLS) and RTMP, UDP, RTP protocols. Some notes:

- RTMP streams should be configured as network inputs to localhost (eg, rtmp://localhost/live/streamname).
- HLS inputs can include options for special handling. For example, including the bandwidth, retries and retry interval progressively downloaded while transcoding. Complete the URI in the style:

  http://<web server>[:port]/path/file.m3u8[?bandwidth=20000&retries=10&retry_interval=2])

  In this example, if the file pointed to by the URI is a variant playlist, the highest bitrate stream will be chosen as the source; the optional bitrate argument can be used to select a specific stream from the playlist. To specify starting n segments from the end, follow this example:

  http://<web server>[:port]/path/file.m3u8[?buffer_segments=n])

  n must be a positive number.
- When using Amazon S3 to host the HLS source content, use this format:

  http(s)://s3-<region>.amazonaws.com/<bucketname>/<object>?<options>

  For example:

  https://s3-eu-west-1.amazonaws.com/elemental.test.ireland2/input/hls.m3u8?retries=40&retry_interval=2

  Specify the S3 Access Key as the username and the Secret Access Key as the password.

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
name	string		Optional name for input. Should be a unique identifier per event.
input_settings	Input Settings	device_input, network_input, smpte2110_input, smpte2022_dash7_network_input, router_input, file_input	Input settings, must be one of device_input, network_input, smpte2110_input, smpte2022_dash7_network_input, router_input, or file_input. Note: replace *input_settings* with the type of input settings you are using in the XML tag (e.g. <network_input>).
order	integer	> 0	Required for multiple inputs. Specifies the order the input should be listed in.
program_id	integer		Selects a specific program from within a multi-program transport stream. If the program doesnâ€™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.
loop_source	boolean	true or **false**	Loop input if it is a file. This allows a file input to be streamed indefinitely.
hot_backup_pair	boolean	true or **false**	Pairs the input with the next listed input to simultaneously run for redundancy, with only one producing an output at a time. Failover conditions dictate input switching, and failback rules determine when to failback upon the resolution.
error_clear_time	integer	At least 1 second longer than the longest failover duration	When a Hot Backup input has failedover, the time threshold to be error free of all failover conditions before failback.
failback_rule	string	Manually, **Immediately**, When Necessary	When a Hot Backup input has been error free for the error clear time duration, the timing for restoring the input to a publishing state.
failover_condition	failover_condition		Input failover condition parameters.
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.
deblock_selected	boolean	**true** or false	Allow the deblock filter when filtering.
denoise_selected	boolean	true or **false**	Allow the denoise 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.
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.

NAME	TYPE	RANGE	DESCRIPTION
prefer_smpte2038	boolean		For a TS containing TR01, specifies whether Live should ingest specific data from a SMPTE-2038 stream in the program or from non-SMPTE-2038 “other” sources. Common examples of ancillary data that Live can ingest are captions, embedded timecodes, and active format descriptors (AFD). If your input is a TS (transport stream) containing TR01, then you typically set this field to true (checked). If your input is a TS without TR01 or is not a TS, then the value of the field is not important. For more information on this field and on TR01 in a transport stream, go to the KBA section of the User Community and search for â€œTS TR01â€?

DEVICE INPUT

NAME	TYPE	RANGE	DESCRIPTION
device_id	integer		
device_type	string		
device_number	integer	**1**	
channel	integer	**1**	Input channel of HD-SDI input card. Currently supports channel 1.
channel_type	string	HD-SDI, HD-2SI, ASI, HDMI, FEC	The input channel type.
*device_channel*_settings	Device Channel Settings	fec_settings, hdmi_settings, sdi_settings	Device channel settings required by the specified channel type. Note: replace *device_channel* with the channel type you are using in the XML tag. For HD-SDI or HD-2SI channel types, the settings tag must be “sdi_settings”. ASI channel types do not require a settings tag.
name	string		The user specified alias for the given device. Can be used in REST calls instead of channel, device_number and channel_type specification.
device_name	string		The hardware specified unique alias for the given device. Can be used in REST calls instead of channel, device_number and channel_type specification.

FEC SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
uri	string	Valid URI	URI of RTP or UDP input to ingest. Should contain a hostname and port (Example: rtp://239.255.1.10:5001). Ports 5000 – 5100 are open by default on the Elemental Live system.
udp_igmp_source	string	IP address	Source address for Source Specific Multicast streams.

HDMI SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
input_format	string	**Auto**, 1080i5994, 1080i60, 1080i50, 1080psf2398, 1080psf24, 1080p2398, 1080p24, 1080p25, 1080p2997, 1080p30, 1080p50, 1080p5994, 1080p60, 720p60, 720p50, 720p5994, NTSC, NTSC2398, NTSC16x9, PAL, and PAL16x9	Input video format.

SDI SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
scte104_offset	integer	-120 – 120	A timing offset can be applied in the SCTE-104 application process. The value is this field will cause the SCTE-104 message to be applied as if it had come in the specified number of frames earlier or later. Specify zero to apply messages at the normal time. Specify a positive number to apply messages later than arrival, and a negative number to apply messages earlier than arrival. If the pre-roll time is smaller than the (negative) offset then the message will be applied as soon as possible.

ROUTER INPUT

NAME	TYPE	RANGE	DESCRIPTION
router_ip	string	Valid IP address	IP address of the router.
router_type	string	blackmagic_videohub, harris_panacea, miranda_nvision	Designates the router type.
*router*_settings	Router Settings	harris_panacea_settings‘, miranda_nvision_settings’	Router settings required by the specified router type. Note: replace *router* with the router type you are using in the XML tag.
input_number	integer	>= 1	Desired SDI input from the router.
input_number_end	integer	>= 1	Currently set automatically to input_number + 3 when quad is set to true. Designates the end of the input number range when using Quadrant 4K input.
quad	boolean		Set when using Quadrant 4K. Determines input range for HD-SDI numbers.
input_format	string	**Auto**, 1080i5994, 1080i60, 1080i50, 1080psf2398, 1080psf24, 1080p2398, 1080p24, 1080p25, 1080p2997, 1080p30, 1080p50, 1080p5994, 1080p60, 720p60, 720p50, 720p5994, NTSC, NTSC2398, NTSC16x9, PAL, and PAL16x9	Input video format.
scte104_offset	integer	-120 – 120	A timing offset can be applied in the SCTE-104 application process. The value is this field will cause the SCTE-104 message to be applied as if it had come in the specified number of frames earlier or later. Specify zero to apply messages at the normal time. Specify a positive number to apply messages later than arrival, and a negative number to apply messages earlier than arrival. If the pre-roll time is smaller than the (negative) offset then the message will be applied as soon as possible.
level_id	integer		

NETWORK INPUT

NAME	TYPE	RANGE	DESCRIPTION
uri	string	Valid URI	URI of RTMP, UDP, HLS input or RTP input to ingest. – For UDP or RTP, specify a hostname and port. Example: udp://239.255.1.10:5001. Make sure that port is open on your appliance. – For HLS, specify http or https as the protocol and specify the location of the manifest. For example, https://203.0.113.0/newschannel/anytownusa.m3u8. – For RTMP inputs specify an address in this format: rtmp://localhost/live/streamname, where “live” is an example of the application name and “streamname” is the application instance. Keep in mind RTMP inputs are pushed to the appliance. In a Conductor cluster, set up RTMP inputs on the primary and backup nodes. (To ingest MPTS over RTP, using the RTP protocol.)
username	string		Username if credentials are required to access file.
password	string		Password if credentials are required to access file.
check_server_certificate	boolean	**true** or false	Check HTTPS server certificates. When disabled, we will still check the cryptography in the certificate, but we will not validate the server’s name. Certain subdomains (notably S3 buckets that use dots in the bucket name) do not strictly match the corresponding certificate’s wildcard pattern and would otherwise cause the event to error. This setting is ignored for other protocols.
interface	string	IP address or name	Optionally specify the network interface to use. Can be entered as the interface IP address (“10.11.12.13”) or as an interface name (“eth2” or “bond0.45”). If left blank, the system routing table will be used to select an interface.
udp_igmp_source	string	IP address or URL.	Source address for Source Specific Multicast streams.
quad	boolean	true or **false**	Configures the program_id into 4 quadrants to support 4K multicast program inputs.
enable fec decode	boolean		Enables SMPTE 2022-1 and SMPTE 2022-2 (ProMPEG) FEC reception on input stream. If FEC data is not received, input will function, but an error will be logged. Only compatible with RTP inputs.
hls_input_settings	HLS Input Settings		Specifies an HLS network input.

SMPTE 2022-7 NETWORK INPUT

NAME	TYPE	RANGE	DESCRIPTION
uri	string	Valid URI	URI of RTP input to ingest. – Specify a hostname and port. Example: rtp://239.255.1.10:5001. Make sure that port is open on your appliance.
interface	string	IP address or name	Optionally specify the network interface to use. Can be entered as the interface IP address (“10.11.12.13”) or as an interface name (“eth2” or “bond0.45”). If left blank, the system routing table will be used to select an interface.
udp_igmp_source	string	IP address or URL.	Source address for Source Specific Multicast streams.
uri_second_stream	string	Valid URI	URI of RTP input to ingest. – Specify a hostname and port. Example: rtp://239.255.1.10:5002. Make sure that port is open on your appliance.
interface_second_stream	string	IP address or name	Optionally specify the network interface to use. Can be entered as the interface IP address (“10.11.12.13”) or as an interface name (“eth2” or “bond0.45”). If left blank, the system routing table will be used to select an interface.
udp_igmp_source_second_stream	string	IP address or URL.	Source address for Source Specific Multicast streams.

FILE INPUT

NAME	TYPE	RANGE	DESCRIPTION
uri	string	Valid file location	File to ingest. Must be accessible to the Elemental Live system, either on the local disk or a SMB mount. File input is decoded in realtime for transcoding and streaming.
hls_input_settings	HLS Input Settings		Specifies an HLS file input.
check_server_certificate	boolean	**true** or false	Check HTTPS server certificates. When disabled, we will still check the cryptography in the certificate, but we will not validate the server's name. Certain subdomains (notably S3 buckets that use dots in the bucket name) do not strictly match the corresponding certificate's wildcard pattern and would otherwise cause the event to error. This setting is ignored for other protocols.

HLS INPUT SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
bandwidth	integer	>= 0 or nil	When specified the HLS stream with the m3u8 BANDWIDTH that most closely matches this value will be chosen, otherwise the highest bandwidth stream in the m3u8 will be chosen. The bitrate is specified in bits per second, as in an HLS manifest.
retries	integer	>= 0 or nil (Default: **10**)	The number of consecutive times that attempts to read a manifest or segment must fail before the input is considered unavailable.
retry_interval	integer	>= 0 or nil (Default: **2**)	The number of seconds between retries when an attempt to read a manifest or segment fails.
buffer_segments	integer	>= 0 or nil	When specified, reading of the HLS input will begin this many buffer segments from the end (most recently written segment). When not specified, the HLS input will begin with the first segment specified in the m3u8.
static_decryption_key	string	Either nil or exactly 32 hexadecimal characters	The 32-character hexadecimal static decryption key for decrypting HLS segments that are AES-128 encrypted.

SMPTE2110 INPUT

NAME	TYPE	RANGE	DESCRIPTION
video_sdp	Video SDP		Specifies a Video SDP input.
audio_sdps	Audio SDP		Specifies an Audio SDP input.
ancillary_sdps	Ancillary SDP		Specifies a Ancillary SDP input.

SDP

NAME	TYPE	RANGE	DESCRIPTION
uri	string		Specify the URI path to the SDP file. Example https://10.10.10.2/sdp_file.sdp
media_index	integer		Specify the index of the media to use from the SDP file. Counts from top down, 0 to (n-1) for n media descriptions in the SDP file.
interface	string		Specify the network interface to use. Example “eth2” or “bond0.45”.

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, sdr_2020, 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. Select 601 to enable transfer function and color gamut per ITU BT.601-7. Select 709 to enable transfer function and color gamut per ITU BT.709-6. Select SDR 2020 to enable transfer function and color gamut per ITU BT.2020-2. Select HDR10 to enable PQ transfer function and non-constant luminance color gamut per ITU BT.2100-1. Select HLG 2020 to enable HLG transfer function and non-constant luminance color gamut per ITU BT.2100-1.
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 is sometimes 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.
default_afd	integer	0 – 15	This four bit AFD value will be applied to frames which have no AFD value. This will only affect video streams which have 'respond to afd' set to 'passthrough' or 'respond'

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.

NAME	TYPE	RANGE	DESCRIPTION
default_selection	boolean	true or **false**	When an Audio Description specifies an audio source and no matching AudioSelector or AudioSelectorGroup is found in the input, then the audio selector marked as "default" will be used. If none are marked as default, silence will be inserted for the duration of the input.
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 channel pair to include in this selector by entering the track ID of the pair. Each pair of audio channels is mapped to one track (channels 1 and 2 = track 1, channels 3 and 4 = track 2, and so on). To combine several channel pairs, enter a comma-separated list of tracks, e.g. "1,2,3" for tracks 1-3 (channels 1-6). To select only one mono channel from a pair, you must select the pair (the track) and audio remix to mute the unwanted mono channel.
offset	integer	integer	Milliseconds to offset the audio from the video. 0 means no offset. A positive number shifts the timestamp of the audio later in time relative to the timestamp of the video. A negative number shifts it earlier relative to the timestamp of the video. For non-file video inputs, the range is -49 to infinite. For a file video input, there are no range restrictions.
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, â€˜PID must be present in inputâ€™ will prevent the event from starting, and â€˜PID may be missing from inputâ€™ 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.
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, Null	Select the style of the captions. You must specify the style: the decoder cannot auto-detect the caption style. See the information on “Supported Caption Formats – Input and Output” in the Working with Captions guide on the Elemental User Community. https://community.elementaltechnologies.com/docs/DOC-1125
***source*_settings**	Source Settings	ancillary_source_settings, embedded_source_settings, file_source_settings, teletext_source_settings, dvb_sub_source_settings, scte27_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 â€˜sccâ€™, â€˜ttmlâ€™, â€˜dfxpâ€™, â€˜stlâ€™, â€˜srtâ€™, and â€˜smiâ€™.
time_delta	integer		Specifies a time delta in seconds to offset the captions from the source file.
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		100 to 8FF	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. Option available for SDI, or for network/file sources when “Prefer SMPTE2038” is selected
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.

INPUT CLIPPING

NAME	TYPE	RANGE	DESCRIPTION
start_timecode	string	valid timecode	Specifies the timecode at which video processing should begin. The given value is interpreted using the timecode source configured for the Live event, which should be set to either embedded or zero-based when using input clipping. Timecode values must be of the format NN:NN:NN:NN, and are interpreted as hours:minutes:seconds:frames. An optional suffix can be provided, specifying the base frame rate the timecode is relative to. When a base frame rate is specified, Live normalizes the given timecode from that base to the input’s frame rate. A sample timecode including the optional suffix is 01:02:03:04/59.94. Suffix values are restricted to the following set {23.976, 24, 25, 29.97, 30, 50, 59.94, 60}. The start_timecode field’s suffix takes precedence over the end_timecode field’s suffix. When no end_timecode is specified, Live process all frames from the configured start_timecode to the end of the media.

NAME	TYPE	RANGE	DESCRIPTION
end_timecode	string	valid timecode	Specifies the timecode at which video processing should end. The given value is interpreted using the timecode source configured for the Live event, which should be set to either embedded or zero-based when using input clipping. Timecode values must be of the format NN:NN:NN:NN, and are interpreted as hours:minutes:seconds:frames. An optional suffix can be provided, specifying the base frame rate the timecode is relative to. When a base frame rate is specified, Live normalizes the given timecode from that base to the input's frame rate. A sample timecode including the optional suffix is 01:02:03:04/59.94. Suffix values are restricted to the following set {23.976, 24, 25, 29.97, 30, 50, 59.94, 60}. The start_timecode field's suffix takes precedence over the end_timecode field's suffix. When no start_timecode is specified, Live will process all frames from the beginning of the media to the configured end_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.png*.

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 – 7	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. May be in timecode (HH:MM:SS:FF) or ISO 8601 UTC Timestamp (20150102T030405.678Z) 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**, systemclock, systemclock_local, zerobased, specifiedstart, reference	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). **System Clock (systemclock)**: Use the UTC time. **Local System Clock (systemclock_local)**: Use the UTC time, adjusted for the timezone specified on the hardware unit where Elemental Live is running. **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. **External Reference Connector (reference)**: Use the time in the external LTC source. Choose this option only if you are using SDI as an input source and have a timecode source connected to your AJA card. If there is more than one AJA card on the hardware unit, the first card is used.
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.
jam_sync_enable	boolean		When checked, timecode is only synced at time window specified by Jam Sync Timecode.
jam_sync_timecode	string	valid timecode	Specifies the time of day at which a jam sync is expected. The time must be of the format NN:NN:NN:NN with values hour:minute:second:frame. During the jam sync window, if the difference is greater than 1 frame and less than 15 frames then a discontinuity will be introduced into the output timecode to resynchronize it with the input timecode. Jam sync does not happen if no value provided.

NIELSEN CONFIGURATION

NAME	TYPE	RANGE	DESCRIPTION
enabled	boolean		Enables Nielsen PCM to ID3 tagging
distributor_id	string		Distribution ID. Enter the Distribution ID assigned to your organization by Nielsen

FAILOVER CONDITION

NAME	TYPE	RANGE	DESCRIPTION
order	integer		Optional parameter used to define order of failover conditions. Use if you wish to explicitly specify the order, otherwise natural order will be used.
description	string	audio_silence, continuity_counter_error, fec_input_correction, **input_loss**, rtp_packet_loss, transport_error_indicator, video_loss, video_black, video_freeze	The type of condition to prompt a failover.
duration	integer	input loss > 100; video loss > 500; audio_silence >= 1000; video_black > 1000; video_freeze > 1000	The amount of time in milliseconds for a failover condition to exist prior to failover. The minimum time for the input loss condition is 100 ms. The minimum time for the video loss condition is 500 ms. The minimum time for the audio silence condition is 1000ms. The minimum time for the video black condition is 1000 ms. The minimum time for the video freeze condition is 1000 ms.
threshold	float	0.0 to 1.0	A decimal value in the range [0.0, 1.0]. This value is used to calculate a pixel difference threshold for black and freeze detection. For Black detection the brightest pixel must be below the threshold for the input to be in the “black” state. For Freeze detection the largest corresponding pixel difference between two frames must be lower than the threshold to be in the “frozen” state. Black Detect Example: SDI 10-bit video input (0×000 to 0×3FF), Black detect threshold set to 0.1 (pixel value of 102.3 = 102). All pixels come in at value 0×040 (64 decimal), so the threshold is triggered. Frozen Detect Example: SDI 10-bit video input (0×000 to 0×3FF), Frozen detect threshold set to 0.01 (pixel difference value of 10.23 = 10). The largest frame to frame pixel differences are approximately 532, so the threshold is NOT triggered. At a later point, the SDI input truly freezes, and the largest frame to frame pixel difference falls to 0. The condition is now triggered.
selector_order	Audio Selector	> 0	Specifies a particular audio stream within an input source. An input may have multiple audio selectors.
error_seconds	integer	1-3600	The number of occurrences of one or more of this error type in a second across “Window Seconds” time period.
window_seconds	integer	1-3600	The time period across which the number of Error Seconds occurs to determine if the failover condition has been met.

FAILURE RULE

NAME	TYPE	RANGE	DESCRIPTION
priority	integer	1 – 100 (Default: **50**)	Failover priority for this Live Event. Higher priority will fill available nodes first. 100 is highest priority.
backup_rule		‘all’, ‘none’, a backup group name, node_id, or [node_id,device_id]	Nodes or Backup Group in a Conductor cluster to be used as backup for this Live Event. Nodes should have the same inputs for proper failover.
restart_on_failure	boolean	true or **false**	Restart this Live Event automatically if it has an error. For instance, if a connection to a CDN times out after the allotted retries.
backoff_time	integer	**30** seconds	Number of seconds to wait until restarting. This is multiplied by the number of previous failures to prevent rapid restarts.
max_failures	integer	**3**	Maximum number of times to restart on failure.

PROCESSORS

NOTIFICATION

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

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. Live Event status XML will be POSTed to this address when the selected events occur.
on_started	boolean	true or **false**	Send notification when Live Event starts.
on_complete	boolean	true or **false**	Send notification when Live Event is stopped.
on_error	boolean	true or **false**	Send notification when Live Event encounters an error.
on_warning	boolean	true or **false**	Send notification when Live Event encounters a warning.
on_alert	boolean	true or **false**	Send notification when an alert occurs while Live Event is running. For example, when an output stream drops below realtime an alert will be issued.
on_clear	boolean	true or **false**	Send notification when an alert is cleared.

PRE-PROCESS

NAME	TYPE	RANGE	DESCRIPTION
script	Location		Script to run before Live Event starts.

POST-PROCESS

NAME	TYPE	RANGE	DESCRIPTION
script	Location		Script to run after the Live Event completes.

TIMING

NAME	TYPE	RANGE	DESCRIPTION
start_type	string	**start_at**, manual	Indicates whether this timing has a scheduled start point, or will be started manually. If manual, an end_type of end_at or duration must be set.
start_at	datetime		Date and time to start. This value is required if start_type is set to start_at.
end_type	string	**end_at**, duration, manual	Indicates whether this timing has a scheduled end point or not. If the end_type is set to end_at (using “On” from the interface), then the end_at parameter is required. If the end_type is set to duration (using “After” from the interface), then the duration parameter is required.
end_at	datetime		Date and time to end. This value is required if end_type is set to end_at.
duration	integer	> 0	The number of minutes to remain active. This value is required if end_type is set to duration, and will manually calculate the end_at parameter given the start_at parameter.

AVAIL BLANKING

NAME	TYPE	RANGE	DESCRIPTION
enabled	boolean	true or **false**	Indicates video, audio and captions will be blanked when insertion metadata is added.
avail_blanking_image	Location		Blanking image to be used. Leave empty for solid black. Only bmp and png images are supported.

BLACKOUT SLATE

NAME	TYPE	RANGE	DESCRIPTION
enabled	boolean	true or **false**	Indicates video, audio and captions will be blanked when indicated by program metadata.
blackout_slate_image	Location		Blackout slate image to be used. Leave empty for solid black. Only bmp and png images are supported.
enable_network_end_blackout	boolean	true or **false**	Enabling this causes the encoder to blackout the video, audio, and captions, and raise the "Network Blackout Image" slate when an SCTE104/35 Network End Segmentation Descriptor is encountered. The blackout will be lifted when the Network Start Segmentation Descriptor is encountered. The Network End and Network Start descriptors must contain a network ID that matches the value entered in "Network ID".
network_id	string	nil	Provides Network ID that matches EIDR ID format (e.g., "10.XXXX/XXXX-XXXX-XXXX-XXXX-XXXX-C").
network_end_blackout_image	string	nil	Path to local file to use as Network End Blackout image. Image will be scaled to fill the entire output raster.

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.
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.
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.

OUTPUT LOCKING

NAME	TYPE	RANGE	DESCRIPTION
enabled	boolean	true or **false**	Indicates Live Event will be enabled for output locking with another encoder.

NAME	TYPE	RANGE	DESCRIPTION
epoch_locking	string	disabled or enabled	This feature is a way to guarantee that multiple events can produce video outputs that are frame-accurate with each other.
protocol	string	multicast or unicast	This feature is a way to guarantee that multiple events can produce video outputs that are frame-accurate with each other. For this feature to work, all the events must communicate with each other via a network address. Specify whether the address is multicast (in which case you can lock any number of events) or unicast (in which case you can lock together only two events).

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.

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.png*.

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 – 7	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.

NAME	TYPE	RANGE	DESCRIPTION
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. May be in timecode (HH:MM:SS:FF) or ISO 8601 UTC Timestamp (20150102T030405.678Z) 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.
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

OUTPUT GROUP

NAME	TYPE	RANGE	DESCRIPTION
type	string	archive_group_settings, apple_live_group_settings, ms_smooth_group_settings, rtmp_group_settings, udp_group_settings, reliable_ts_group_settings, dash_iso_group_settings, smpte_st2110_group_settings	Output group settings type – defines the type of this output group.
name	string		
order	integer	> 0	Required for multiple output groups. Specifies the order the output groups should be listed in.
***output_group*_settings**	Group Settings	archive_group_settings, apple_live_group_settings, dash_iso_group_settings, ms_smooth_group_settings, rtmp_group_settings, udp_group_settings, reliable_ts_group_settings, smpte_st2110_group_settings	Output group type-specific settings. Note: replace *output_group* with the group type you are using in the XML tag.
output	Output		Output settings. There can be multiple outputs within a group.
external output	External Output		External outputs are outputs sourced from an external location. Can be used to define backup streams in meta .m3u8 playlist. Only valid for Apple HLS output groups with generate_meta_file enabled.
custom_name	string		Custom group name to be defined by user. Only letters, numbers and the underscore character allowed; only 16 characters allowed.

ARCHIVE 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	> 1	Number of seconds to write to archive file before closing and starting a new one. Leave blank to disable archive rollover.

APPLE LIVE GROUP SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
destination	Location		A directory 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. See Uri Types for supported protocols. If the schema is S3, query parameters endpoint and region may be included to explicitly define the S3 endpoint and region.
interface	string	IP address or name	Optionally specify the network interface to use. Can be entered as the interface IP address (“10.11.12.13”) or as an interface name (“eth2” or “bond0.45”). If left blank, the system routing table will be used to select an interface.
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.

NAME	TYPE	RANGE	DESCRIPTION
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.
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.
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.
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. Playback of VOD mode content during event is not guaranteed due to HTTP server caching problems.
keep_segments	integer	>= 1 (Default: **21**)	Number of segments to retain in the destination directory. vod_mode must be false for this setting to have an effect.
index_n_segments	integer	>= 1 (Default: **10**)	Number of segments to keep in the playlist (.m3u8) file. vod_mode must be false for this setting to have an effect, and this number should be less than or equal to keep_segments.
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.
timestamp_delta_milliseconds	integer		Provides an extra millisecond delta offset to fine tune the timestamps.
cdn	string	None, Basic_PUT, Akamai, WebDAV, AWS Elemental MediaStore	Type of HTTP communication to use for pushing to origin server: 1) None – not allowed for HTTP destinations 2) Akamai – compatibility with Akamai CDN inputs 3) Basic_PUT – No creation of folders but deletes old files based on the setting in "Keep Segments" field. 4) WebDAV – HTTP PUT, PROPFIND, MKCOL, DELETE. Creates folders as needed and deletes old files based on the settings in "Keep Segments" field. 5) AWS Elemental MediaStore – HTTPS PUT. Creates folders as needed and deletes old files based on the settings in "Keep Segments" field. Enter the Access Key ID in the Username field. Enter the Secret Access Key in the Password field.
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**)	Number of retry attempts that will be made before the Live Event is put into an error state.
filecache_duration	integer	0 – 600	Size in seconds of file cache for streaming outputs.

NAME	TYPE	RANGE	DESCRIPTION
restart_delay	integer	>= 0 (Default: 15)	If a streaming output fails, number of seconds to wait until a restart is initiated. A value of 0 means never restart.
token	string		Token parameter for authenticated Akamai. If not specified, *gda* is used.
salt	string		Salt for authenticated Akamai.
log_uploads	boolean	true or **false**	Log uploads to CDN in a file called /opt/elemental_se/web/log/job_<id>/upload_<groupid>.csv. A row is generated for each file POST (created) and each file DELETE.
chunked_transfer	boolean	true or **false**	Use chunked transfer encoding to Basic_PUT, WebDAV, Akamai or AWS Elemental MediaStore. User should contact Akamai to enable this feature.
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_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.
encryption_type	string	AES-128, SAMPLE-AES	Encrypts the segments with the given encryption scheme. Leave blank to disable. Selecting 'Disabled' in the web interface also disables encryption.
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.

NAME	TYPE	RANGE	DESCRIPTION
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, speke	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.
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, daterange	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.
on_input_loss	string	**emit_output**, pause_output	Specifies how the output should be handled if input is lost. Pause Output (pause_output): Follow the behavior for the repeat field in the Input Loss Behavior set of fields. Then, if that time expires, stop output for this output group. Emit Content (emit_output): Follow the behavior controlled by the repeat, black, and color/slate fields in the Input Loss Behavior set of fields. With this option, Live continues to produce output.
send_endlist_tag	boolean		If true, the EXT-X-DLIST tag will be sent when the Live event stops. If false, the EXT-X-DLIST tag will not be sent and the last partial segment will not be published.

MICROSOFT SMOOTH STREAMING GROUP SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
publish_point	Location		A directory and basename to save as an .ismv archive. Alternately, an HTTP destination for streaming to a Smooth publishing point. If the base filename is left blank, the base filename of the first input will be inserted.
check_server_certificates	boolean	**true** or true	Verify the https certificate chain to a trusted Certificate Authority (CA). This will cause https outputs to self-signed certificates to fail unless those certificates are manually added to the OS trusted keystore.
interface	string	IP address or name	Optionally specify the network interface to use. Can be entered as the interface IP address (“10.11.12.13”) or as an interface name (“eth2” or “bond0.45”). If left blank, the system routing table will be used to select an interface.

NAME	TYPE	RANGE	DESCRIPTION
fragment_length	integer	>= 1 (Default: **2**)	Length of mp4 fragments to generate (in seconds). Fragment length must be compatible with GOP size and framerate.
connection_retry_interval	integer	Default: **2**	Number of seconds to wait before retrying connection to the IIS server if the connection is lost. Content will be cached during this time and the cache will be be delivered to the IIS server once the connection is re-established.
num_retries	integer	Default: **10**	Number of retry attempts.
use_event_id	boolean	**true** or false	If true, the specified or generated event ID will be passed to the IIS Server. If false and the same Live Event is used without changing the publishing point, clients might see cached video from the previous run.
event_id	string		Event ID to pass to the IIS server. If not set, a timestamp will be used for this value, so restarts will not replay cached content. If set, content will be appended to the specified event id for the given publish point across Live Event restarts.
send_eos	boolean	**true** or false	If true, the end of stream signal will be sent to the IIS Server when the Live Event stops. If false, the end of stream signal will not be sent.
send_stream_manifest	boolean	**true** or false	Send stream manifest so publishing point doesn't start until all streams start.
timestamp_offset_today	boolean		Offset Smooth Streaming timestamps from midnight on the day the Live Event started. NOTE: This field depends on the timecode_passthrough parameter in individual streams that make up this output group. When timecode_passthrough is false, Smooth Streaming timestamps will start at 0 and this parameter has no effect. When timecode_passthrough is true, timestamps will start at the Live Event's timecode source initial value. A value of systemclock is recommended for timecode source.
timestamp_offset	date		Start date to offset Smooth Streaming timestamps. Only applies if timestamp_offset_today is false. NOTE: This field depends on the timecode_passthrough parameter in individual streams that make up this output group. When timecode_passthrough is false, Smooth Streaming timestamps will start at 0 and this parameter has no effect. When timecode_passthrough is true, timestamps will start at the Live Event's timecode source initial value plus this offset. A value of systemclock is recommended for timecode source.
timestamp_delta_seconds	integer		Additional seconds of offset to be added to the timestamp_offset date.
timestamp_delta_milliseconds	integer		Additional milliseconds of offset to be added to the timestamp_offset date.
enable_sparse_track	boolean	nil	Use incoming SCTE-35 messages to generate a sparse track in this group of MS-Smooth outputs.
acquisition_point_id	string	nil	The value of the "Acquisition Point Identity" element used in each message placed in the sparse track. Only enabled if "Enable Sparse Track" is checked.
collapse_identical_audio_streams	boolean		When checked, audio streams with the same settings are removed from all but one of the video streams.
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.

NAME	TYPE	RANGE	DESCRIPTION
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.
filecache_duration	integer	0 – 600	Size in seconds of file cache for streaming outputs.
restart_delay	integer	>= 0 (Default: 15)	Number of seconds before initiating a restart due to output failure, due to exhausting the num_retries on one segment, or exceeding filecache_duration.
log_uploads	boolean	true or **false**	Log uploads to CDN in a file called /opt/elemental_se/web/log/ job_<id>/upload_<groupid>.csv. A row is generated for each file POST (created) and each file DELETE.
key_provider_settings	Key Provider Settings	irdeto_settings, seachange_settings, conax_settings, piksel_settings, inside_secure_settings	Key Provider-specific settings.
send_delay_ms	integer	0..10000	Outputs that are "output locked" can use this delay. Assign a delay to the output that is "secondary". Do not assign a delay to the "primary" output. The delay means that the primary output will always reach the downstream system before the secondary, which helps ensure that the downstream system always uses the primary output. (If there were no delay, the downstream system might flip-flop between whichever output happens to arrive first.) If the primary fails, the downstream system will switch to the secondary output. When the primary is restarted, the downstream system will switch back to the primary (because once again it is always arriving first)
on_input_loss	string	**emit_output**, pause_output	Specifies how the output should be handled if input is lost. Pause Output (pause_output): Follow the behavior for the repeat field in the Input Loss Behavior set of fields. Then, if that time expires, stop output for this output group. Emit Content (emit_output): Follow the behavior controlled by the repeat, black, and color/slate fields in the Input Loss Behavior set of fields. With this option, Live continues to produce output.

DASH ISO GROUP SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
destination	Location		A directory or HTTP destination for DASH streaming, with no file extension. When Media Content Destination is specified, only MPD and initialization segment are sent here. See Uri Types for supported protocols.
interface	string	IP address or name	Optionally specify the network interface to use. Can be entered as the interface IP address ("10.11.12.13") or as an interface name ("eth2" or "bond0.45"). If left blank, the system routing table will be used to select an interface.
chunked_encoding_enabled	boolean	true or **false**	When selected, Chunk Length may be configured
chunk_length_milliseconds	integer	>= 200 and <= 500 (Default: **200**)	Length of chunks to generate (in milliseconds). This value will be rounded to the nearest video frame boundary, such that chunk length in frames evenly divides fragment length in frames.
fragment_length	integer	>= 1 (Default: **2**)	Length of fragments to generate (in seconds). This value will be rounded to the nearest video frame boundary, such that fragment length measured in frames evenly divides GOP size. Hence, actual fragment length may be longer. Fragment length will be an integral multiple of chunk length.
segment_length	integer	>= 1 (Default: **30**)	Length of MPD segments to create (in seconds). This value will be rounded to the nearest video frame boundary, such that segment length measured in frames an integral multiple of fragment length.

NAME	TYPE	RANGE	DESCRIPTION
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.
vod_mode	boolean	true or **false**	If checked, generates single output file that contains all segments. If not checked, generates individual files of ‘Segment Length’ and deletes segments older than ‘Keep Segments’.
keep_segments	integer	>= 1 (Default: **21**)	Number of segments to retain in the destination directory. vod_mode must be false for this setting to have an effect.
index_n_segments	integer	>= 1 (Default: **10**)	Number of segments to keep in the manifest (.mpd) file. vod_mode must be false for this setting to have an effect, and this number should be less than or equal to keep_segments.
use_subdirectories	boolean	true or **false**	Place segments in subdirectories. vod_mode must be false for this setting to have an effect.
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.
live_to_vod	boolean	false or **true**	When live event ends, convert the manifest (.mpd) file to a static type with an event duration. vod_mode must be false for this setting to have an effect.
hbbtv_enabled	boolean	true or **false**	Supports HbbTV specification version 1.5
cdn	string	None, Akamai, Basic_PUT, WebDAV, AWS Elemental MediaStore	Type of HTTP communication to use for pushing to origin server: 1) None – not allowed for HTTP destinations 2) Akamai – compatibility with Akamai CDN inputs 3) Basic_PUT – No creation of folders but deletes old files based on the setting in “Keep Segments” field. 4) WebDAV – HTTP PUT, PROPFIND, MKCOL, DELETE. Creates folders as needed and deletes old files based on the settings in “Keep Segments” field. 5) AWS Elemental MediaStore – HTTPS PUT. Creates folders as needed and deletes old files based on the settings in “Keep Segments” field. Enter the Access Key ID in the Username field. Enter the Secret Access Key in the Password field.
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**)	
media_content_destination	Location		When specified, media segments are sent to this destination, separating from MPD and initialization segment. See Uri Types for supported protocols.
media_content_cdn	string	None, Akamai, Basic_PUT, WebDAV, AWS Elemental MediaStore	Type of HTTP communication to use for pushing to origin server: 1) None – not allowed for HTTP destinations 2) Akamai – compatibility with Akamai CDN inputs 3) Basic_PUT – No creation of folders but deletes old files based on the setting in “Keep Segments” field. 4) WebDAV – HTTP PUT, PROPFIND, MKCOL, DELETE. Creates folders as needed and deletes old files based on the settings in “Keep Segments” field. 5) AWS Elemental MediaStore – HTTPS PUT. Creates folders as needed and deletes old files based on the settings in “Keep Segments” field. Enter the Access Key ID in the Username field. Enter the Secret Access Key in the Password field.
media_content_connection_retry_interval	integer	>= 0 (Default: **2**)	Number of seconds to wait before retrying connection to the CDN if the connection is lost.
media_content_num_retries	integer	>= 0 (Default: **10**)	
media_content_chunked_transfer	boolean	true or **false**	Use chunked transfer encoding to Basic_PUT, WebDAV, Akamai or AWS Elemental MediaStore. User should contact Akamai to enable this feature.
filecache_duration	integer	0 – 600	Size in seconds of file cache for streaming outputs.

NAME	TYPE	RANGE	DESCRIPTION
restart_delay	integer	>= 0	If a streaming output fails, number of seconds to wait until a restart is initiated. A value of 0 means never restart.
log_uploads	boolean	true or **false**	Log uploads to CDN in a file called /opt/elemental_se/web/log/job_<id>/upload_<groupid>.csv. A row is generated for each file POST (created) and each file DELETE.
chunked_transfer	boolean	true or **false**	Use chunked transfer encoding to Basic_PUT, WebDAV, Akamai or AWS Elemental MediaStore. User should contact Akamai to enable this feature.
minimum_update_period	integer	>= 0	Smallest period between potential changes to the MPD.
min_buffer_time	integer	>= 0	Minimum time of initially buffered media that is needed to ensure smooth playout.
suggested_presentation_delay	integer	> 0	When not set in UI, maximum of 20s and 2x Segment Length and minBufferTime is used.
media_available_time	datetime		This updates availabilityStartTime tag in DASH MPD, indicating the live content will be available at the specified time. If not provided, availabilityStartTime is set to the earliest availability time for the first media segment. Format is yyyy-MM-ddTHH:mm:ssZ for UTC. When Z is not included, timecode input is treated as local time and converted to UTC.
drm_system	string	**nil**, widevine, playready, or multidrm	Specifies DRM system used for DASH outputs. None by default.
key_provider_settings	Key Provider Settings	piksel_settings, generic_cenc_settings, playready_cenc_settings, speke_settings	Key Provider-specific settings. For DASH ISO, this is constrained by the DRM System selection.
key_rotation_count	integer		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.

ADOBE RTMP GROUP SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
cdn	string	**None**, Akamai, Limelight, Level3, Edgecast, Internap, Wowza	CDN to authenticate for this group of Adobe RTMP outputs.
cache_length	integer	>= 30	Cache length, in seconds, is used to calculate buffer size.
restart_delay	integer	>= 0 (Default: 15)	If a streaming output fails, number of seconds to wait until a restart is initiated. A value of 0 means never restart.
disconnect_immediately	boolean	**true** or false	Controls behavior when content cache fills up. If remote origin server stalls the RTMP connection and does not accept content fast enough the ‘Media Cache’ will fill up. When the cache reaches the length specified in ‘Cache Length’ the cache will stop accepting new content. If this box is checked, the RTMP output will force a disconnect. Clear the media cache, and reconnect after ‘Restart Delay’ seconds. If the box is unchecked, the RTMP output will wait up to 5 minutes to allow the origin server to begin accepting data again.
ad_markers	string	onAkamaiAdPod, onCuePoint, onCuePointMSec, onCuePointSCTE35, onUserDataEvent	Choose one or more ad marker types to pass SCTE35 signals through to this group of Adobe RTMP outputs.

NAME	TYPE	RANGE	DESCRIPTION
caption_data	string	all, 608_field1_and_field2, 608_field1	Controls the types of data that passes to OnCaption outputs. If set to 'all' then 608 and 708 carried DTVCC data will be passed. If set to '608_field1_and_field2' then DTVCC data will be stripped out, but 608 data from both fields will be passed. If set to '608_field1' then only the data carried in 608 from field 1 video will be passed.
enable_oncuepoint_broadcast_time	boolean	true or **false**	When enabled the encoder will transmit an OnCuePoint message every 5 seconds with a timestamp of the specific encoding time. The encoding time is denoted in seconds since the UNIX Epoch (Jan 1st, 1970).
broadcast_time_offset	integer		Offset in milliseconds to apply to enable_oncuepoint_broadcast_time.
onfi_timecode_frequency	integer	>= 0 (Default: 1)	onFI timecode output frequency. Must be set to multiples of GOP size. Default is 1 x GOP. 0 means no onFI.

UDP GROUP SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
name	string		UDP or RTP output groups require a name and no other parameters; this is a known issue that will be addressed in a future release.
on_input_loss	string	drop_program, **drop_ts**, emit_program	Specifies how the output should be handled if input is lost. Stop transmitting TS (drop_ts) : Default. Follow the behavior for the repeat field in the Input Loss Behavior set of fields. Then, if that time expires, stop the transport stream. Emit Content (emit_program): With this option, Live continues to produce output. The video PID for this program will contain content based on the behavior controlled by the repeat, black, and color/slate fields in the Input Loss Behavior set of fields. The audio PID will consist of silence. There will be no PID for the captions. Drop program from TS (drop_program): Follow the behavior for the repeat field in the Input Loss Behavior set of fields. Then, if that time expires, omit this program from the transport stream. Omit reference to this program from the PAT. Omit the PMT for this program. Omit the PIDs for this program. The transport stream will consist of the PAT and the NULL PID.
csp_settings	Csp Settings		The StreamID emitted in the Cross Streaming Prevention messages.
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.

RELIABLE TS GROUP SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
name	string		A Reliable TS output group requires a name.
on_input_loss	string	drop_program, **drop_ts**, emit_program	Specifies behavior of last resort when Input Video is lost, and no more backup inputs are available. The Entire Transport Stream can be shut down, or the program can be dropped from the stream (and replaced with Null Packets to meet TS bitrate requirement), or the last good picture can be repeated indefinitely.

CSP SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
stream_id	string		The StreamID emitted in the Cross Streaming Prevention messages. A URI string for segmentation_upid().
Interval	integer	100 – 10000	The period between Cross Stream Prevention message insertions. Units are in milliseconds.

SMPTE ST 2110 GROUP SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
on_input_loss	string	**drop_ts**, emit_program	Specifies how the output should be handled if input is lost. 1) Stop transmitting TS (drop_ts): Default. Follow the behavior for the repeat field in the Input Loss Behavior set of fields. Then, if that time expires, stop the transport stream. 2) Emit Content (emit_program): With this option, Live continues to produce output. The video stream will contain content based on the behavior controlled by the repeat, black, and color/slate fields in the Input Loss Behavior set of fields. Any audio stream will consist of silence. The ancillary data stream will be empty.
enable_nmos	string	false **true**	Manage output via NMOS API specifications IS-04 (Discovery & Registration) and IS-05 (Device Connection Management). When the Live event is started the output will be registered on the network as an NMOS device and each of its streams will be registered as an NMOS sender. Each stream destination can be configured through the connection management portion of the NMOS web service.

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, AWS Elemental MediaStore	Type of HTTP communication to use for pushing to origin server: 1) None – not allowed for HTTP destinations 2) Akamai – compatibility with Akamai CDN inputs 3) Basic_PUT – No creation of folders but deletes old files based on the setting in “Keep Segments” field. 4) WebDAV – HTTP PUT, PROPFIND, MKCOL, DELETE. Creates folders as needed and deletes old files based on the settings in “Keep Segments” field. 5) AWS Elemental MediaStore – HTTPS PUT. Creates folders as needed and deletes old files based on the settings in “Keep Segments” field. Enter the Access Key ID in the Username field. Enter the Secret Access Key in the Password field.
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**)	

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.

NAME	TYPE	RANGE	DESCRIPTION
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.
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 â€œ0xâ€? 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 Live Event 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 and DASH ISO 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	mp4, f4v, raw, m2ts, m3u8, ismv, rtmp, mov, uvu, 3gp, mxf, 2110	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, rtmp_settings, udp_settings, reliable_ts_settings, smpte_st2110_output_settings	Specific settings for this type of output. Required for outputs within Adobe RTMP, UDP or Reliable TS group.
*container*_settings	Container Settings	mov_settings, m2ts_settings, raw_settings, mp4_settings, f4v_settings, smpte_st2110_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. For HLS outputs SCTE-35 is automatically enabled when ad markers associated with the Apple HLS group are enabled.
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.
smpte_2038	boolean	true or **false**	Enables passthrough of non-audio SDI ADPs to output TS, per SMPTE 2038 standard.
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.
timed_metadata_passthrough	boolean	true or **false**	Enables passthrough of timed metadata from input to output.
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.

NAME	TYPE	RANGE	DESCRIPTION
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.
insert_private_metadata	boolean	true or **false**	If true, inserts private metadata from the private_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.
arib_captions_passthrough	boolean	true or **false**	If true, passes any ARIB Captions data from the input source to this output. Only available for certain containers under certain conditions.
insert_amf_metadata	boolean		If true, inserts metadata in AMF format from the amf_metadata REST command into RTMP output, on specified Ad Marker.
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. (This is ignored when generating DOlby Vision output.)

APPLE LIVE SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
segment_type	string	TS or fMP4	Specify if this output should produce TS or fMP4 segments.
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

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

ADOBE RTMP SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
rtmp_endpoint	Location		Endpoint for Flash Media Server. For connection to Akamai, a username and password must be supplied in rtmp_endpoint. Endpoint URI fields accept Format Identifiers.
stream_name	string		Stream name for Flash Media Server. Accepts Format Identifiers.
connection_retry_interval	integer	Default: **2**	Number of seconds to wait before retrying a connection to the Flash Media server if the connection is lost.
num_retries	integer	Default: **10**	Number of retry attempts.

UDP SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
destination	Location		Applies only if mpts_membership is none or remote. Destination address and port number for Transport Stream packets. Can be unicast or multicast. If mpts_membership is none, can be UDP or RTP. If mpts_membership is remote and secondary_destination will not be specified, can be UDP or RTP. If mpts_membership is remote and secondary_destination will be specified, can only be RTP. For example, udp://239.10.10.10:5001 or rtp://10.100.100.100:5002.
secondary_destination	Location		Applies only if mpts_membership is none or remote. Additional destination address and port number for Transport Stream packets. Can be unicast or multicast. If mpts_membership is none, can be UDP or RTP; If mpts_membership is remote, can only be RTP. For example, udp://239.10.10.10:5001 or rtp://10.100.100.100:5002.
interface	string	IP address or name	Optionally specify the network interface to use. Can be entered as the interface IP address (“10.11.12.13”) or as an interface name (“eth2” or “bond0.45”). If left blank, the system routing table will be used to select an interface.

NAME	TYPE	RANGE	DESCRIPTION
buffer_msec	integer	0 – 10000 (10 seconds)	UDP output buffering in milliseconds. Larger values increase latency through the transcoder but simultaneously assist the transcoder in maintaining a constant, low-jitter UDP/RTP output while accommodating clock recovery, input switching, input disruptions, picture reordering, etc.
mpts_membership	string	**None**, Local, Remote	Specifies the location of the multiplexer used with this encode. If “None”, no multiplexing will occur. If “Local” the onboard MPTS process will be used and all communication will be automatically handled on local interfaces. If “Remote”, the Live Encoder will send data to a remote multiplexer using the specified addresses.
complexity_transmit_destination	string		Multicast destination for complexity estimates from encoder to statmux. Only required for use with Elemental Statmux (separate multiplexer).
secondary_complexity_transmit_destination	string		Optional. Additional multicast destination for complexity estimates from encoder to statmux. Only required for use with Elemental Statmux (separate multiplexer). Must be received by the same statmux as the Primary Complexity Transmit Destination. Typical use case is to enable multiple network path redundancy (two interfaces and two switches) from encoder to statmux.
allocation_receipt_destination	string		Multicast destination for bitrate allocations from statmux to encoder. Only required for use with Elemental Statmux (separate multiplexer).
secondary_allocation_receipt_destination	string		Optional. Additional multicast destination for bitrate allocations from statmux to encoder. Only required for use with Elemental Statmux (separate multiplexer). Must be received by the same encoder as the Primary Bitrate Allocation Destination. Typical use case is to enable multiple network path redundancy (two interfaces and two switches) from statmux to encoder.
max_ts_packet_count	integer	1 – 7	Sets the number of MPEG TS packets to be sent in each IP packet. Lower values slightly reduce latency, at the cost of network overhead.

RELIABLE TS SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
Delivery Protocol	string	mediaconnect or Zixi	Indicates the type of Reliable TS protocol to carry the Transport Stream. Current options are ‘AWS Elemental MediaConnect’ and ‘Zixi’.
Destination/Amazon Resource Name	string		Destination specification for a Reliable Transport Stream output. For an output of type AWS Elemental MediaConnect, the destination is an AWS Elemental MediaConnect Flow ARN. User credentials required. For an output of type Zixi, the destination is an IP address:port number and must start with the prefix: zixi:// For example: zixi://10.11.42.42:2088. A Stream ID is required. User credentials are not required.
interface	string	IP address or name	Optionally specify the network interface to use. Can be entered as the interface IP address (“10.11.12.13”) or as an interface name (“eth2” or “bond0.45”). If left blank, the system routing table will be used to select an interface.

SMPTE ST 2110 OUTPUT SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
destination	Location		Can be unicast or multicast. Must be rtp and must include a port. For example, rtp://239.10.10.10:5001 or rtp://10.100.100.100:5002.
secondary_destination	Location		Can be unicast or multicast. Must be rtp and must include a port. For example, rtp://239.10.10.10:5001 or rtp://10.100.100.100:5002.

NAME	TYPE	RANGE	DESCRIPTION
interface	string	IP address or name	Specify the network interface to use. Can be entered as the interface IP address (“10.11.12.13”) or as an interface name (“eth5” or “bond0.45”). Must be configured to the Mellanox card.
interlace_mode	enum	**progressive**, top_field, bottom_field	This is only valid on outputs containing video streams.
packet_time_microseconds	integer	125, **1000**	Length of time per packet in microseconds. This is only valid on outputs containing audio streams.
par_numerator	integer		Pixel Aspect Ratio numerator. If Pixel Aspect Ratio is Follow source, can be left blank: defaults to aspect ratio of source. For other Pixel Aspect Ratio options, must be specified.
par_denominator	integer		Pixel Aspect Ratio denominator. If Pixel Aspect Ratio is Follow source, can be left blank: defaults to aspect ratio of source. For other Pixel Aspect Ratio options, must be specified.
par_follow_source	boolean	**true** or false	No pixel aspect ratio conversion from source.

FEC OUTPUT SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
include_column_fec	boolean	**true**	Enables column-based FEC; must be enabled.
include_row_fec	boolean	**true** or false	Enables row-based FEC; enabled by default.
column_depth	integer	4-20	Parameter D from SMPTE 2022-1. The height of the FEC protection matrix. The number of transport stream packets per column error correction packet. Must be between 4 and 20, inclusive
row_length	integer	1-20	Parameter L from SMPTE 2022-1. The width of the FEC protection matrix. Must be between 1 and 20, inclusive. If only Column FEC is used, then larger values increase robustness. If Row FEC is used, then this is the number of transport stream packets per row error correction packet, and the value must be between 4 and 20 inclusive.

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

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.
cc_in_pmt	boolean	true or **false**	When enabled, generates caption_service_descriptor in PMT.
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.
iframe_only	boolean	true or **false**	When true, output only key frames while dropping all other frames. VBR mode will be enabled and Buffer Model will not be employed.
dvb	boolean	true or **false**	When true, the stream type is 0×06 for AC3 and EAC3 audio; both the DVB ac3_audio_descriptor, and the DVB AU_information data fields will be included; and the output uses the DVB buffer model for Dolby Digital audio. When false, the stream type is 0×81 for AC3 audio; the stream type is 0×87 for EAC3 audio; the ATSC audio_descriptor data field will be included; and the output uses the ATSC buffer model for Dolby Digital audio.
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.
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 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.
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.
drop_absent_streams	boolean	true or **false**	When checked (true) output audio streams will be removed from the program if the selected input audio stream is removed from the input. This allows the output audio configuration to dynamically change based on input configuration. If this box is not checked (false), all output audio streams will output encoded silence when not connected to an active input stream.
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	string	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	string	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	string	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	string	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	string	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.
scte27_pids	string	32 – 8182 (Default: **450-459**)	Packet Identifier (PID) for input source SCTE-27 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.

NAME	TYPE	RANGE	DESCRIPTION
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. Enter “detect” to have PID detected from the input.
scte35_esam_pid	string	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.
smpte_2038_output_pid	string	32 – 8182 (Default: **509**)	PID for SMPTE 2038 data.
klv_data_pids	string	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	string	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.
private_metadata_pid	string	32 – 8182 (Default: **503**)	Packet Identifier (PID) of the private metadata stream in the transport stream. Can be entered as a decimal or hexadecimal value.
etv_platform_pid	string	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	string	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	string	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.
arib_captions_pid	string	32 – 8182	Packet Identifier (PID) for ARIB Captions in the transport stream. Can be entered as a decimal or hexadecimal value.
zixi_latency	integer	0 – 16000 milliseconds	The Zixi MAX LATENCY value. This defines the maximum latency allowed by the Zixi protocol.
zixi_streamid	string		Optional Zixi Stream ID
zixi_encryption	string		The Zixi Encryption level. Valid values are: NO_ENCRYPTION, AES-128, AES-192, AES-256
zixi_keyvalue	string		A Zixi Key Value is a string of hexadecimal values which coincide with the specified Zixi encryption level. Valid values for each encryption type are as follows: AES-128: 32 character string of 16 hexadecimal bytes AES-192: 48 character string of 24 hexadecimal bytes AES-256: 64 character string of 32 hexadecimal bytes

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
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	string	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	string	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	string	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	string	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.
private_metadata_pid	string	32 – 8182 (Default: **503**)	Packet Identifier (PID) of the private 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.

NAME	TYPE	RANGE	DESCRIPTION
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	string	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	string		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.

SMPTE ST 2110 SETTINGS

This currently doesn’t have any specific settings, but needs to be included in the XML as <smpte_st2110_settings/>

NAME	TYPE	RANGE	DESCRIPTION

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.

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 g external outputs are ordered together.
external uri	string		The external output feature lets you include a reference to the variant manifest from another output group (on this or another Li master manifest for this output group. For example, this output group produces curling.m3u8, curling_high.m3u8, and the corre and publishes them to the destination https://example.com/sports. Another output group produces backup_curling.m3u8, backu and the corresponding .ts segments and publishes them to the destination https://example2.com/sports. You can set up so that on https://example.com/sports includes references to curling_high.m3u8 (as usual) and to backup_curling_high.m3u8. Enter th manifest. For example, https://example2.com/backup_curling.m3u8. (Or, if the destination is the same for both output groups, e backup_curling.m3u8)
stream_attributes	string		Enter the information to include in the master manifest, in the EXT-X-STREAM-INF line for this variant. BANDWIDTH is require specification, other fields are optional and depend on what your downstream systems expect. For example, BANDWIDTH=690800,AVERAGE-BANDWIDTH=690800,CODECS=“avc1.4d4015,mp4a.40.2”,RESOLUTION=480×270,AUDI To ensure compliance with the HLS standard, be careful with the syntax for example, note the lack of spaces, note the use of a double quotes around lists and strings.

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 Live Event 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 or name must be provided, specifying by permalink is not supported.

VIDEO DESCRIPTION

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

NAME	TYPE	RANGE	DESCRIPTION
codec	enum	h.264, 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 strength of the anti-alias filter used for scaling. 0 is the softest setting, 100 is the sharpest. A setting of 50 is 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 â€˜trueâ€™ 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. â€™Falseâ€™ 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.
selected_gpu	string	**Auto (blank)**, comma separated list of values 0,1,2,3	GPUs to encode this stream on. Leaving this blank will cause Elemental Live to decide the best GPU. If the operator runs multiple Live Events at the same time, they can manually balance the Live Event GPUs for optimal performance. Streams with video using the HEVC (h.265) codec can designate two or more GPUs, or if left to auto, additional GPUs will be assigned automatically. GPUs excluded at the system settings will be ignored.
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.
y	integer		Top of rectangle.
width	integer		Width of rectangle in pixels.
height	integer		Height of rectangle in pixels.

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.
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, Statmux, QVBR	Rate control mode. CQ uses constant quantizer (qp), ABR (average bitrate) does not write HRD parameters. Statmux allows for statistical multiplexing on outputs with an MPTS Membership. QVBR: Sets a bitrate that meets the desired quality (specified in the Quality Level field). The bit rate will not exceed Max Bitrate and will not fall below the bitrate required to meet the desired quality.
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. Five megabits can be entered as 5000000 or 5m. Five hundred kilobits can be entered as 500000 or 0.5m. Required for QVBR.
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.
quality_level	float	1.0 – 10.0 (Default: **7.0**)	Target quality value in steps of 1/3. Applicable only to QVBR mode. 1.0 is the lowest quality and 10.0 is the highest and approaches lossless. Typical levels for content distribution are between 6.0 and 8.0.
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. If blank, value is Bitrate x 2.
min_buf_occ	integer	>=0 to buf_size	Minimum occupancy of VBV / HRD buffer in bits. If blank, value is 0.
buf_fill_pct	integer	0 – 100	Percentage of the buffer that should initially be filled (HRD buffer model). If blank, value is 90.
framerate_numerator	integer		Framerate numerator – framerate is a fraction, e.g. 24000 / 1001 = 23.976 fps. If Framerate is Follow source, can be left blank: defaults to framerate of source. For other Framerate options, must be specified.
framerate_denominator	integer		Framerate denominator. If Framerate is Follow source, can be left blank: defaults to framerate of source. For other Framerate options, must be specified.
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. - Hard Telecine is only valid with interlace modes of “Top Field First” and “Bottom Field First”; Soft Telecine is only valid with the “Progressive” interlace mode.
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.

NAME	TYPE	RANGE	DESCRIPTION
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 â€œtop field firstâ€? and â€œbottom field firstâ€?. - If the source is progressive, the output will be interlaced with â€œtop field firstâ€? or â€œbottom field firstâ€? 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.
progressive_references	boolean	true or **false**	Adjust position of P and B frames within a GOP so that progressive-scan reference frames are used whenever possible. Improves compression efficiency of mixed progressive & interlace content, particularly hard telecine codec film content.
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**)	Applies only when scd_mode is On or Transition Detection. In a stream that belongs to an output group that is defining an ABR stack, always set this field to 0. In a stream that is not part of an ABR stack, enter a value that forces a minimum separation between repeated (cadence) I-frames and I-frames inserted by scene change detection (SCD). Enter the value as a number of frames. - If an SCD I-frame is within the specified interval before a cadence I-frame, then the SCD I-frame is inserted but the planned cadence I-frame is not inserted. The current GOP is shrunk. The normal GOP cadence then resumes. - If an SCD I-frame is within the specified interval after a cadence I-frame, then the planned cadence I-frame is not inserted and instead the current GOP is stretched to the SCD I-frame. The normal GOP cadence then resumes. The maximum GOP stretch = GOP size + Min-I-interval â€“ 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. If blank, feature is off.
qp	integer	1 – 51	Quantization parameter – fixed for CQ rate control mode, or starting QP for rate controller. If blank, field is ignored.
max_qp	integer	1 – 51	Maximum QP for rate controller. If blank, field is ignored.

NAME	TYPE	RANGE	DESCRIPTION
min_qp	integer	1 – 51	Minimum QP for rate controller. If blank, field is ignored.
par_follow_source	boolean	**true** or false	No pixel aspect ratio conversion from source.
par_numerator	integer		Pixel Aspect Ratio numerator. If Pixel Aspect Ratio is Follow source, can be left blank: defaults to aspect ratio of source. For other Pixel Aspect Ratio options, must be specified.
par_denominator	integer		Pixel Aspect Ratio denominator. If Pixel Aspect Ratio is Follow source, can be left blank: defaults to aspect ratio of source. For other Pixel Aspect Ratio options, must be specified.
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_mode	enum	off, **on** or transition_detection	**On:** inserts I-frames when scene change is detected. **Off:** does not force an I-frame when scene change is detected. **Transition Detection:** recommended when Rate Control Mode (rate_control_mode) is QVBR. In a stream that belongs to an output group that is defining an ABR stack, set all streams to On or Transition Detection, or set all streams to Off.
look_ahead_rate_control	string	low, **medium**, high	Amount of lookahead. A value of low can decrease latency and memory usage, while high can produce better quality for certain content.
num_ref_frames	integer	**1** – 6	Minimum number of reference frames to use. The actual number of reference frames used by the encoder for a given event is the **maximum** of the following numbers: â€¢ The number of Reference Frames specified in the event â€¢ If GOP Reference B-Frame is true and Motion Vector Direct Mode is Spatial: 4 â€¢ If GOP Reference B-Frame is true and Motion Vector Direct Mode is not present or is not Spatial: 5 â€¢ If B Frames is 0 and Interlace Mode is Progressive: 1 â€¢ If B Frames is 0 and Interlace Mode is not Progressive: 2 â€¢ If B Frames is not 0: 2
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.
gop_markers	boolean	true or **false**	Inserts a Recovery Point SEI message for open GOPs, or starts a new sequence for closed GOPs.
dynamic_sub_gop	boolean	true or **false**	Adjust number of b-frames per sub-GOP based on motion, up to maximum specified for ‘B Frames’. Higher motion uses fewer b-frames. Improves subjective video quality for high-motion content.
gop_b_reference	boolean	true or **false**	Enable use of reference B frames for GOP structures that have B frames > 1.
svq	integer	-3.0 Higher Quality, -2.0, -1.0, **0.0**, 1.0, 2.0, 3.0: Higher Density	Selects encoding features based on performance. Higher values use fewer system resources so may allow more streams to be encoded. NOTE. a value of -3.0 is NOT available on GPU encodes.
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.
passes	integer	**1** or 2	Number of encoding passes.

NAME	TYPE	RANGE	DESCRIPTION
motion_vector_direct_mode	string		Specifies the mode to use for the B-frame's Direct mode motion-vector prediction. Applies only when the codec is H.264 (AVC) and a CPU encoder is applied, or when the codec is H.264 and a GPU encoder is used and Density vs Quality (svq) is set to â€œ-2.0 or -3.0: Higher Qualityâ€?. For any other scenario, the Direct mode motion vector type is always "temporal". Default is â€œspatialâ€?. Spatial Direct mode is recommended in general. Temporal Direct mode may provide better subjective video quality when there are more clean linear motions. Auto mode applies either spatial or temporal prediction, depending on which one gives better predicted motion vectors for the given B-frame.

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, Statmux, QVBR	Rate control mode. CQ uses constant quantizer (qp), ABR (average bitrate) does not write HRD parameters. Statmux allows for statistical multiplexing on outputs with an MPTS Membership. QVBR: Sets a bitrate that meets the desired quality (specified in the Quality Level field). The bit rate will not exceed Max Bitrate and will not fall below the bitrate required to meet the desired quality.
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. Five megabits can be entered as 5000000 or 5m. Five hundred kilobits can be entered as 500000 or 0.5m. Required for QVBR.
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.
quality_level	float	1.0 – 10.0 (Default: **7.0**)	Target quality value in steps of 1/3. Applicable only to QVBR mode. 1.0 is the lowest quality and 10.0 is the highest and approaches lossless. Typical levels for content distribution are between 6.0 and 8.0.
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. If blank, value is Bitrate x 2.
buf_fill_pct	integer	0 – 100	Percentage of the buffer that should initially be filled (HRD buffer model). If blank, value is 90.

NAME	TYPE	RANGE	DESCRIPTION
min_i_interval	integer	0 – 30 (Default: **0**)	Applies only when scd_mode is On or Transition Detection. In a stream that belongs to an output group that is defining an ABR stack, always set this field to 0. In a stream that is not part of an ABR stack, enter a value that forces a minimum separation between repeated (cadence) I-frames and I-frames inserted by scene change detection (SCD). Enter the value as a number of frames. - If an SCD I-frame is within the specified interval before a cadence I-frame, then the SCD I-frame is inserted but the planned cadence I-frame is not inserted. The current GOP is shrunk. The normal GOP cadence then resumes. - If an SCD I-frame is within the specified interval after a cadence I-frame, then the planned cadence I-frame is not inserted and instead the current GOP is stretched to the SCD I-frame. The normal GOP cadence then resumes. The maximum GOP stretch = GOP size + Min-I-interval â€“ 1.
framerate_numerator	integer		Framerate numerator – framerate is a fraction, e.g. 24000 / 1001 = 23.976 fps. If Framerate is Follow source, can be left blank: defaults to framerate of source. For other Framerate options, must be specified.
framerate_denominator	integer		Framerate denominator. If Framerate is Follow source, can be left blank: defaults to framerate of source. For other Framerate options, must be specified.
framerate_follow_source	boolean	**true** or false	No framerate conversion from source.
num_ref_frames	integer	**1** – 6	Minimum number of reference frames to use. The actual number of reference frames used by the encoder for a given event is the **maximum** of the following numbers: â€¢ The number of Reference Frames specified in the event â€¢ If GOP Reference B-Frame is true and Motion Vector Direct Mode is Spatial: 4 â€¢ If GOP Reference B-Frame is true and Motion Vector Direct Mode is not present or is not Spatial: 5 â€¢ If B Frames is 0 and Interlace Mode is Progressive: 1 â€¢ If B Frames is 0 and Interlace Mode is not Progressive: 2 â€¢ If B Frames is not 0: 2
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.
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.
dynamic_sub_gop	boolean	true or **false**	Adjust number of b-frames per sub-GOP based on motion, up to maximum specified for ‘B Frames’. Higher motion uses fewer b-frames. Improves subjective video quality for high-motion content.
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. If blank, field is ignored.
min_qp	integer	1 – 51	Minimum QP for rate controller. If blank, field is ignored.
max_qp	integer	1 – 51	Maximum QP for rate controller. If blank, field is ignored.

NAME	TYPE	RANGE	DESCRIPTION
par_follow_source	boolean	**true** or false	No pixel aspect ratio conversion from source.
par_numerator	integer		Pixel Aspect Ratio numerator. If Pixel Aspect Ratio is Follow source, can be left blank: defaults to aspect ratio of source. For other Pixel Aspect Ratio options, must be specified.
par_denominator	integer		Pixel Aspect Ratio denominator. If Pixel Aspect Ratio is Follow source, can be left blank: defaults to aspect ratio of source. For other Pixel Aspect Ratio options, must be specified.
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	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_mode	enum	off, **on** or transition_detection	**On:** inserts I-frames when scene change is detected. **Off:** does not force an I-frame when scene change is detected. **Transition Detection:** recommended when Rate Control Mode (rate_control_mode) is QVBR. In a stream that belongs to an output group that is defining an ABR stack, set all streams to On or Transition Detection, or set all streams to Off.
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.
svq	float	-3.0: Higher Quality, -2.0, -1.0, **0.0**, 0.5, 1.0, 2.0, 3.0: 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	**default**, adaptive, off	Specify Sample Adaptive Offset (SAO) filter strength. Adaptive mode dynamically selects best strength based on content
sei_timecode	boolean	true or **false**	Inserts timecode for each frame as 4 bytes of an unregistered SEI message.
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 â€œtop field firstâ€? and â€œbottom field firstâ€?. - If the source is progressive, the output will be interlaced with â€œtop field firstâ€? or â€œbottom field firstâ€? 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. - Hard Telecine is only valid with interlace modes of “Top Field First” and “Bottom Field First”; Soft Telecine is only valid with the “Progressive” interlace mode.
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.

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, Statmux, QVBR	Rate control mode. CQ uses constant quantizer (qp), ABR (average bitrate) does not write HRD parameters. Statmux allows for statistical multiplexing on outputs with an MPTS Membership. QVBR: Sets a bitrate that meets the desired quality (specified in the Quality Level field). The bit rate will not exceed Max Bitrate and will not fall below the bitrate required to meet the desired quality.
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. Applicable only to VBR and QVBR modes. 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. If blank, value is Bitrate x 2.
buf_fill_pct	integer	0 – 100	Percentage of the buffer that should initially be filled (HRD buffer model). If blank, value is 90.
framerate_numerator	integer		Framerate numerator – framerate is a fraction, e.g. 24000 / 1001 = 23.976 fps. If Framerate is Follow source, can be left blank: defaults to framerate of source. For other Framerate options, must be specified.
framerate_denominator	integer		Framerate denominator. If Framerate is Follow source, can be left blank: defaults to framerate of source. For other Framerate options, must be specified.

NAME	TYPE	RANGE	DESCRIPTION
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.
dynamic_sub_gop	boolean	true or **false**	Adjust number of b-frames per sub-GOP based on motion, up to maximum specified for ‘B Frames’. Higher motion uses fewer b-frames. Improves subjective video quality for high-motion content.
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 â€œtop field firstâ€? and â€œbottom field firstâ€?. - If the source is progressive, the output will be interlaced with â€œtop field firstâ€? or â€œbottom field firstâ€? 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.
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.
progressive_references	boolean	true or **false**	Adjust position of P and B frames within a GOP so that progressive-scan reference frames are used whenever possible. Improves compression efficiency of mixed progressive & interlace content, particularly hard telecine codec film content.

NAME	TYPE	RANGE	DESCRIPTION
min_i_interval	integer	0 – 30 (Default: **0**)	Applies only when scd_mode is On or Transition Detection. In a stream that belongs to an output group that is defining an ABR stack, always set this field to 0. In a stream that is not part of an ABR stack, enter a value that forces a minimum separation between repeated (cadence) I-frames and I-frames inserted by scene change detection (SCD). Enter the value as a number of frames. - If an SCD I-frame is within the specified interval before a cadence I-frame, then the SCD I-frame is inserted but the planned cadence I-frame is not inserted. The current GOP is shrunk. The normal GOP cadence then resumes. - If an SCD I-frame is within the specified interval after a cadence I-frame, then the planned cadence I-frame is not inserted and instead the current GOP is stretched to the SCD I-frame. The normal GOP cadence then resumes. The maximum GOP stretch = GOP size + Min-I-interval â€“ 1.
adaptive_quantization	string	off, low, **medium**, high	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. If blank, feature is off.
qp	integer	1 – 112	Quantization parameter – fixed for CQ rate control mode, or starting QP for rate controller. If blank, field is ignored.
max_qp	integer	1 – 112	Maximum QP for rate controller. If blank, field is ignored.
min_qp	integer	1 – 112	Minimum QP for rate controller. If blank, field is ignored.
par_numerator	integer		Pixel Aspect Ratio numerator. If Pixel Aspect Ratio is Follow source, can be left blank: defaults to aspect ratio of source. For other Pixel Aspect Ratio options, must be specified.
par_denominator	integer		Pixel Aspect Ratio denominator. If Pixel Aspect Ratio is Follow source, can be left blank: defaults to aspect ratio of source. For other Pixel Aspect Ratio options, must be specified.
par_follow_source	boolean	**true** or false	No pixel aspect ratio conversion from source.
scd_mode	enum	off, **on** or transition_detection	**On:** inserts I-frames when scene change is detected. **Off:** does not force an I-frame when scene change is detected. **Transition Detection:** recommended when Rate Control Mode (rate_control_mode) is QVBR. In a stream that belongs to an output group that is defining an ABR stack, set all streams to On or Transition Detection, or set all streams to Off.
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).
min_buf_occ	integer	>=0 to buf_size	Minimum occupancy of VBV / HRD buffer in bits. If blank, value is 0.

PRORES SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
framerate_numerator	integer		Framerate numerator – framerate is a fraction, e.g. 24000 / 1001 = 23.976 fps. If Framerate is Follow source, can be left blank: defaults to framerate of source. For other Framerate options, must be specified.

NAME	TYPE	RANGE	DESCRIPTION
framerate_denominator	integer		Framerate denominator. If Framerate is Follow source, can be left blank: defaults to framerate of source. For other Framerate options, must be specified.
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. If Pixel Aspect Ratio is Follow source, can be left blank: defaults to aspect ratio of source. For other Pixel Aspect Ratio options, must be specified.
par_denominator	integer		Pixel Aspect Ratio denominator. If Pixel Aspect Ratio is Follow source, can be left blank: defaults to aspect ratio of source. For other Pixel Aspect Ratio options, must be specified.
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 â€œtop field firstâ€? and â€œbottom field firstâ€?. - If the source is progressive, the output will be interlaced with â€œtop field firstâ€? or â€œbottom field firstâ€? 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.
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.

NAME	TYPE	RANGE	DESCRIPTION
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. If Framerate is Follow source, can be left blank: defaults to framerate of source. For other Framerate options, must be specified.
framerate_denominator	integer		Framerate denominator. If Framerate is Follow source, can be left blank: defaults to framerate of source. For other Framerate options, must be specified.
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	**I420**, NV12, YV12, YV16, YUYV, UYVY, v210, s210	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_sdr_2020, force_hdr10, force_hlg_2020, force_dv_5, force_dv_8_1	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. Select 601 to enable transfer function and color gamut per ITU BT.601-7. Select 709 to enable transfer function and color gamut per ITU BT.709-6. Select SDR 2020 to enable transfer function and color gamut per ITU BT.2020-2. Select HDR10 to enable PQ transfer function and non-constant luminance color gamut per ITU BT.2100-1. Select HLG 2020 to enable HLG transfer function and non-constant luminance color gamut per ITU BT.2100-1. Select Dolby Vision Profile 5 to enable IPT color space and Dolby Vision metadata. Requires Video Range to be Full Swing. Select Dolby Vision Profile 8.1 to enable PQ transfer function and non-constant luminance color gamut per ITU BT.2100-1 and Dolby Vision metadata.

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.png*.

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 – 7	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. May be in timecode (HH:MM:SS:FF) or ISO 8601 UTC Timestamp (20150102T030405.678Z) 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.
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 â€“ 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

The Noise Reducer filters can help output quality if the content will be compressed heavily. To determine the best option, test the different filters on the expected source content.

NAME	TYPE	RANGE	DESCRIPTION
filter	string	**Bilateral**, Mean, Gaussian, Lanczos, Sharpen, Conserve, Spatial, Temporal	â€¢ Mean / Gaussian / Lanczos: All of these algorithms allow for varying blur strengths. Mean is the strongest filter (it operates on a smaller group of pixels), while Lanczos is the mildest (it operates on a larger group of pixels). â€¢ Sharpen: Sharpens the edges instead of softening them. â€¢ Conserve: Limits the pixel values to within the minimum and maximum values of the neighboring pixel values. It is designed to reduce speckle noise or camera sensor noise. It can be useful for old film stock with excessive film grain noise. â€¢ Bilateral: This algorithm preserves strong edges but flattens subtle textures. It is useful for bitrate reduction with some blurring of details. At high strengths it produces a cel-shaded look. â€¢ Spatial: A human perception-based filter that removes input content complexity in the order of noticeability by the human eye. It is more computationally expensive than other filters. It filters pictures adaptively based on frequency and contrast masking, so detail is removed where differences are not perceptible. Lower strengths reduce the complexity of the picture with no perceptible loss of detail. At higher strengths, the image will be visually simplified but the critical detail is preserved. â€¢ Temporal: Uses the same frequency and contrast masking as Spatial, plus motion masking.
strength	integer	0-16 for Spatial and Temporal filters, 0-3 for other filters	Relative strength of filtering (higher produces stronger filtering).
speed	integer	-2 to 3 for Spatial; -1 to 3 for Temporal.	Applies only to the Spatial or Temporal filter. The speed of the filter (higher number is faster). Low setting reduces bit rate at the cost of density, high setting improves density at the cost of bit rate.

NAME	TYPE	RANGE	DESCRIPTION
post_filter_sharpen_strength	integer	0 to 3	Applies only to the Spatial filter. The strength of post-noise-reduction sharpening filter, with 0 disabling the filter and 3 enabling it at maximum strength.
aggressive_mode	integer	0 to 4	Applies only to Temporal filter. The relative strength of motion masking (higher produces stronger filtering). Set Aggressive Mode to higher value when operating at lower CBR bitrates or VBR max bitrates to filter high complexity scenes more aggressively. 0-2 for complexity reduction with minimal sharpness loss; 3-8 for complexity reduction with image preservation; 9-16 for reduced noise combined with high complexity reduction.
temporal_filter_sharpen	boolean	false, true	Default is to apply a sharpen filter after applying the Motion Compensated Temporal Filter. By un-checking this checkbox, the sharpening will NOT be applied.

WATERMARKING

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

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

TIMECODE BURN-IN

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

AUDIO DESCRIPTION

NAME	TYPE	RANGE	DESCRIPTION
codec	enum	aac, mp2, wav, aiff, ac3, ec3, pass through, dtse, pcm	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, pcm_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.

NAME	TYPE	RANGE	DESCRIPTION
language_code	string	IETF-RFC5646 Language Tag code which is a superset of ISO 639-2 three-digit code	Indicates the IETF-RFC5646 Language Tag of the audio output track. This 'Language Code' is composed of one or more subtags, each of which refines or narrows the range of language identified by the overall tag. For the purpose of the Live Software, the IETF-RFC5646 Language Tag is identified as the 'Language Code'. The 'Language Code' entered here will be associated with the specified audio stream when 'Follow Input Language Code' is NOT selected or when 'Follow Input Language Code' is selected but there is NO language code specified by the input.
follow_input_language_code	boolean		Choosing â€˜Follow Input Language Codeâ€™ will cause the 'Language Code' of the output to follow the 'Language Code' of the input. The language specified in the 'Language Code' field will be used when 'Follow Input Language Code' is not selected or when 'Follow Input Language Code' is selected but there is no 'Language Code' specified by the input.
audio_type	integer		Applies only if Follow Input Audio Type is unchecked (false). A number between 0 and 255. The following are defined in ISO-IEC 13818-1: 0 = Undefined, 1 = Clean Effects, 2 = Hearing Impaired, 3 = Visually Impaired Commentary, 4-255 = Reserved.
follow_input_audio_type	boolean		Checked (set to true): If the input contains an ISO 639 audio_type, then that value is passed through to the output. If the input contains no ISO 639 audio_type, the value in Audio Type is included in the output. Unchecked (set to false): The value in Audio Type is included in the output. Note that this field and Audio Type are both ignored if Set Broadcaster Mix Descriptor is checked (true).
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.
arib_dynamic_audio_track	integer	1, 2	When enabled, input audio channel selection and AAC settings will be dynamically updated based on the encoding of 'audio mode' found in ARIB STD B-39 VANC.
timecode_passthrough	boolean		If enabled for an audio-only MS Smooth output, the fragment absolute time will be set to the current timecode. This option does not write timecodes to the audio elementary stream.
nielsen_rtvod_watermark	string	c3, c7	Insert C3/C7 tag into a stream that already has NAES II watermarks present.

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.

NAME	TYPE	RANGE	DESCRIPTION
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.
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.
vbr_quality	enum	LOW1, LOW2, LOW3, MEDIUM1, MEDIUM2, MEDIUM3, HIGH1, HIGH2, HIGH3	VBR Quality Level – Only used if rate_control_mode is VBR.

WAV SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
sample_rate	integer	8000 – 192000 (Default: **44100**)	Sample rate in hz.
channels	integer	1, **2**, 4, 8	Mono, Stereo, 4-Channel, or 8-Channel.
bit_depth	integer	**16** or 24	Bits per sample.

AIFF SETTINGS

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

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.
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.

NAME	TYPE	RANGE	DESCRIPTION
stereo_downmix	string	**not_indicated**, lo_ro, lt_rt, dpl2	Stereo downmix preference. Only used for 3/2 coding mode.
lt_rt_center_mix_level	float	3.0, 1.5, 0.0, -1.5, **-3.0**, -4.5, -6.0, -60	Left total/Right total center mix level. Only used for 3/2 coding mode.
lt_rt_surround_mix_level	float	-1.5, **-3.0**, -4.5, -6.0, -60	Left total/Right total surround mix level. Only used for 3/2 coding mode.
lo_ro_center_mix_level	float	3.0, 1.5, 0.0, -1.5, **-3.0**, -4.5, -6.0, -60	Left only/Right only center mix level. Only used for 3/2 coding mode.
lo_ro_surround_mix_level	float	-1.5, **-3.0**, -4.5, -6.0, -60	Left only/Right only surround mix level. Only used for 3/2 coding mode.
phase_shift_90_degree	boolean	**true** or false	Applies a 90-degree phase shift to the surround channels. Only used for 3/2 coding mode.
attenuate_3_db	boolean	true or **false**	Applies a 3 dB attenuation to the surround channels. Only used for 3/2 coding mode.
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.

PCM SETTINGS

NAME	TYPE	RANGE	DESCRIPTION
channels	integer	1, **2**, 4, 6, 8	Number of audio channels.
bit_depth	integer	16, **24**	Number of bits per sample.
sample_rate	integer	**48000**, 96000	Sample rate in hz.

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

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

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

AUDIO NORMALIZATION SETTINGS

Requires license

NAME	TYPE	RANGE	DESCRIPTION
algorithm	string	**1770-1** or 1770-2	Audio normalization algorithm to use. 1770-1 conforms to the CALM Act specification, 1770-2 conforms to the EBU R-128 specification.
correct_audio	boolean	**true** or false	When enabled the output audio is corrected using the chosen algorithm. If disabled, the audio will be measured but not adjusted.
target_lkfs	float	-59 to 0	Target LKFS to adjust volume to. If no value is entered, a default value will be used according to the chosen algorithm. The CALM Act (1770-1) recommends a target of -24 LKFS. The EBU R-128 specification (1770-2) recommends a target of -23 LKFS.
log_loudness	boolean	**true** or false	Log each output’s audio track loudness to a CSV file.
truepeak	boolean	true or **false**	Calculate and log the TruePeak for each output’s audio track loudness.

CAPTION DESCRIPTION

NAME	TYPE	RANGE	DESCRIPTION
order	integer	> 0	Required for multiple captions. Specifies the order the caption descriptions should be listed in.
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.

NAME	TYPE	RANGE	DESCRIPTION
destination_type	string	ARIB, Burn-In, CFF-TT, DVB-Sub, EBU-TT-D, **Embedded**, Embedded+SCTE-20, SCTE-20+Embedded, RTMP CaptionInfo, RTMP CuePoint, SCC, SMI, SMPTE-TT, SRT, Teletext, TTML, WebVTT, SCTE-27	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	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	Applies when output is TTML, CFF-TT, or EBU-TT-D and source captions are Teletext, TTML, SMPTE-TT, CCF-TT, embedded, or an embedded combination, or when output is WebVTT and source captions are Teletext, embedded, or an embedded combination. (For other input/output combinations, the style is always simplified, which means that the downstream player determines the style.) Check (true) to pass the style information from the source to the output captions. Uncheck (false) to use simplified style.
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).
id3_as_caption_content	boolean	true or **false**	When enabled, ID3 text packets are inserted into the caption track as content. Only available with TTML destinations and MS Smooth outputs.
captions_subtype	string	Exactly four capital letters (A-Z)	When specified, overrides the default “CAPT” value for the subtype attribute in the MS Smooth manifest. Only available with TTML destinations and MS Smooth outputs.
dvb_sub_to_sd	boolean	true or **false**	Check (true) to resize the captions down to SD (720×576). This field is useful when HD caption images cannot be handled by the video player. Uncheck (false) to leave captions in their original size. Applicable only if both the input and output captions are DVB-Sub. Value is ignored for all other combinations.

MS SMOOTH TTML TEMPLATES

There are two template files used by Elemental Live 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, smart	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. Selecting "smart" justification will left-justify live subtitles and center-justify pre-recorded subtitles. 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 inputs and DVB-Sub/Burn-in outputs.
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.

NAME	TYPE	RANGE	DESCRIPTION
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. Specify a framerate that matches the framerate of the associated video. 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 https://server/presets/my_preset. If left blank, a permalink will be generated based on the Preset name.
log_edit_points	boolean	true or **false**	Generates an XML file in the event 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		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	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.
arib_captions_passthrough	boolean	true or **false**	If true, passes any ARIB Captions data from the input source to this output. Only available for certain containers under certain conditions.

NAME	TYPE	RANGE	DESCRIPTION
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.
smpte_2038	boolean	true or **false**	Enables passthrough of non-audio SDI ADPs to output TS, per SMPTE 2038 standard.

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.

LIVE EVENT PROFILE

NAME	TYPE	RANGE	DESCRIPTION
name	string		Name for Profile.
description	string		Description for Profile.
permalink	string	Alphanumeric characters and underscores, cannot be an integer.	A short unique identifier used to refer to this Live Event Profile. For example, if the permalink is “my_profile”, it can be accessed at https://server/live_event_profiles/my_profile. If left blank, a permalink will be generated based on the Live Event Profile name.
input	Input		Input parameters. There can be multiple inputs in a single Profile.
timecode_config	Timecode Config		Contains settings used to acquire and adjust timecode information from inputs.
loop_all_inputs	boolean	true or **false**	Process list of inputs sequentially and loop from the first input when complete.
ad_trigger	string	**scte35_splice_insert**, scte35_time_signal_apos	Controls which types of SCTE signals signal Ad Avails. Ads can be signaled with “Splice Insert” messages, which is traditional, or with “Time Signal” messages, carrying “Ad Placement Opportunity Start” segmentation messages (type_id 0×35). See SCTE 35 2013 for more information.
ignore_web_delivery_allowed_flag	boolean	true or **false**	When enabled, Segment Descriptors with web_delivery_allowed_flag set to 0 will no longer trigger blackouts or Ad Avail slates
ignore_no_regional_blackout_flag	boolean	true or **false**	When enabled, Segment Descriptors with no_regional_blackout_flag set to 0 will no longer trigger blackouts or Ad Avail slates
initial_audio_gain	integer	-60 to 60 dB (Default: **0 dB**)	Value to set the initial audio gain for the Live Event. This is also editable while the Live Event is running.
notification	Notification		Settings for notification on status changes.
avsync_enable	boolean	**true** or false	Enables A/V sync.
avsync_pad_trim_audio	boolean	**true** or false	Enables A/V sync trim audio.
pre_process	Pre-Process		Settings for preprocessing steps.
post_process	Post-Process		Settings for postprocessing steps.

NAME	TYPE	RANGE	DESCRIPTION
failure_rule	Failure Rule		Settings for failure rules.
image_inserter	Image Inserter		Settings for the image inserter. When attached to a profile, inserts images into the decoded input and appears in every output.
avail_blanking	Avail Blanking		Settings for ad avail blanking.
blackout_slate	Blackout Slate		Settings for blackout slate.
output_lock	Output Lock		Settings for output locking.
input_end_action	string	switch_input, or none	Indicates the action to take when an input completes (e.g. end-of-file.) Options include immediately switching to the next sequential input (via "switch_input") or transcoding black / color / slate images per the "Input Loss Behavior" configuration until an activate_input REST command is received (via "none").
output_timing_source	string	input_clock, or system_clock	Indicates whether the rate of frames emitted by the Live encoder should be paced by its system clock (which optionally may be locked to another source via NTP) or should be locked to the clock of the source that is providing the input stream.
input_buffer_size	integer	4 – 300 (Default: **60**)	Number of frames to buffer at input. Higher values will allow less dropped frames, but use more memory. Lower values can improve streaming latency.
resource_reservation	string	none or 4k_decode	When 4K Decode is selected, the system reserves additional resources to provide real-time 4K decode of a second network input. This option allows the system to reserve resources for 4K seamless input switching or 4K Hot-Hot redundancy when network (IP) sources are used. Resource reservation is not required when using Quad SDI sources for 4K encoding. Note – this option only has effect for 4K workflows hosted on the AWS Elemental Live L700AE series (or greater) when 4K encoding is configured. The REST parameter controlled by this checkbox is resource_reservation. It supports values of none and 4k_decode.
low_framerate_input	boolean	true or **false**	Adjusts video input buffer for streams with very low video framerates. This is commonly used for music channels with less than one video frame per second.
low_latency_mode	boolean	true or **false**	Reduces latency of audio/video sync. This reduces overall latency of Live Event, but may result in more dropped audio packets on input timestamp discontinuities. Parameter values such as B Frame and Min-I Interval may still increase latency while Low Latency is set, but the net effect is an overall reduction.
stream_assembly	Stream Assembly		Stream assemblies for this Profile. A Profile can have several stream assemblies which define output codec settings.
nielsen_configuration	Nielsen Configuration		Nielsen configuration settings
output_group	Output Group		Output groups for this Profile. Output groups contain information about where streams should be distributed to.
ad_avail_offset	integer	-1000 – 1000 (Default: **0**)	When specified, this offset (in milliseconds) is added to the input Ad Avail PTS time.
user_data	string		User-defined data to be attached to the Live Event. This data is available with Live Event status requests via the API.
extract_sdt	boolean	true or **false**	Extracts SDT information from input stream. Displays Service Provider and Service Names during running state.

SCHEDULE

NAME	TYPE	RANGE	DESCRIPTION
name	string		Name of the schedule.

NAME	TYPE	RANGE	DESCRIPTION
profile	integer or string	Valid Profile ID or name	The Profile to be used to create the scheduled Live Events. A valid ID or name must be provided, specifying by permalink is not supported.
node	integer or string	Valid node ID or hostname or IP Address	Node on which to run the scheduled Live Events.
failure_rule	Failure Rule		Scheduled Live Event failure parameters.
input	Input		Input parameters to use when creating scheduled Live Events. If the Profile being used has input parameters, then this may be left blank and the input parameters of the Profile will be used.
start_type	string	**start_time**	Indicates which field specifies the scheduled Live Event start times.
start_time	datetime		Date and time to start the first scheduled Live Event. This value is required if start_type is set to start_time.
end_type	string	**end_time**, duration	Indicates which field specifies the scheduled Event end times. If the end_type is set to end_time, then the end_time parameter is required. If the end_type is set to duration, then the duration parameter is required.
end_time	datetime		Date and time to end the first scheduled Live Event. This value is required if end_type is set to end_time.
duration	integer		The number of minutes to remain active. This value is required if end_type is set to duration.
until	string	**forever**, end_date	Indicates how long this schedule is set to repeat. If the value is end_date, then the end_date parameter is required.
end_date	datetime		Date that this repeating schedule ends. Required if the until parameter is set to end_date.
schedule_type	string	**daily**, weekly, monthly	Indicates the type of repeating schedule. Only the parameters that start with <schedule_type> will be followed.
daily_x	integer	**1** – 30	Indicates the number of days on which to repeat. If value is 2, schedule will repeat every 2 days, starting with the day indicated by the start_timing. This value is required if schedule_type parameter is set to daily. Default value is 1, which indicates a schedule that repeats every day.
weekly_sunday	boolean	true or **false**	Indicates that this schedule will repeat every Sunday. Only available if schedule type parameter is set to weekly.
weekly_monday	boolean	true or **false**	Indicates that this schedule will repeat every Monday. Only available if schedule type parameter is set to weekly.
weekly_tuesday	boolean	true or **false**	Indicates that this schedule will repeat every Tuesday. Only available if schedule type parameter is set to weekly.
weekly_wednesday	boolean	true or **false**	Indicates that this schedule will repeat every Wednesday. Only available if schedule type parameter is set to weekly.
weekly_thursday	boolean	true or **false**	Indicates that this schedule will repeat every Thursday. Only available if schedule type parameter is set to weekly.
weekly_friday	boolean	true or **false**	Indicates that this schedule will repeat every Friday. Only available if schedule type parameter is set to weekly.
weekly_saturday	boolean	true or **false**	Indicates that this schedule will repeat every Saturday. Only available if schedule type parameter is set to weekly.
monthly_x	integer	**1** – 12	Indicates the number of months on which to repeat. If value is 2, schedule will repeat every 2 months, starting with the month indicated by the start_timing. This value is required if schedule_type parameter is set to monthly. Default value is 1, which indicates a schedule that repeats every month.
monthly_by	string	**day**, week	Indicates the type of repeating monthly schedule. A value of day requires the monthly_day_of_month parameter to be set. A value of week requires the monthly_day_of_week and the monthly_week_of_month parameters to be set.
monthly_day_of_month	integer	1 – 31	Indicates which day of the month to repeat on. A value of 15 will repeat on the 15th day of the month. Required if monthly_by parameter is set to day.

NAME	TYPE	RANGE	DESCRIPTION
monthly_day_of_week	string	Sunday – Saturday	Indicates which day of the week to repeat on. Combines with monthly_week_of_month to determine which day to repeat on each month. Required if monthly_by parameter is set to week.
monthly_week_of_month	string	First, Second, Third, Fourth, Last	Indicates which week of the month to repeat on. Combines with monthly_day_of_week to determine which day to repeat on each month. Required if monthly_by parameter is set to week. For example, if monthly_day_of_week is Sunday, and monthly_week_of_month is Last, then it will repeat on the Last Sunday of the month.

CUE POINT PARAMETERS

TRIGGERING AN AVAIL

NAME	TYPE	RANGE	DESCRIPTION
event_id	integer		A SCTE-35 event ID is required for this event. This allows for canceling later. Typically, each Avail will have its own unique event ID. If no event ID is given, or if a value of zero is provided, an ID will be generated and returned in the response. Auto generated ID starts from 0xf0000001 and continues to increment. It wraps around to 0xf0000001 after 0xffffffff.
splice_time/hours	integer		The time at which to start the avail. Preroll + Duration should be less than 13 hours 15 minutes.
splice_time/minutes	integer		The minute at which to start the avail.
splice_time/seconds	integer		The second at which to start the avail.
splice_time/frames	integer		The frame (within hour+minute+second) at which to start the avail.
splice_offset	integer		The number of milliseconds to wait (also known as “preroll”) before triggering this avail.
duration	integer		Return from avail after this many seconds.

Either a splice_time (based on the timecode_source of the Live Event) or splice_offset (preroll milliseconds) must be given. Examples:

```
<cue_point>
    <event_id>1</event_id>
    <splice_time>
        <hours>0</hours>
        <minutes>0</minutes>
        <seconds>0</seconds>
        <frames>0</frames>
    </splice_time>
    <duration>30</duration>
</cue_point>
```

```
<cue_point>
    <event_id>2</event_id>
    <splice_offset>8000</splice_offset>
    <duration>30</duration>
</cue_point>
```

Note that cue_point times are in the local timezone when the timecode_source is systemclock. To query for the current time of the stream while the Live Event is running, issue the following command:

```
curl -H "Accept: application/xml" -H "Content-type: application/xml" \
-d"<cue_point><get_current_time>1</get_current_time></cue_point>" \
https://<server_ip>/api/live_events/4/cue_point
```

This command returns a splice point time you can base future cue points on as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
```

```
  <tag>1</tag>
  <splice_time>
    <hours>0</hours>
    <seconds>2</seconds>
    <minutes>0</minutes>
    <frames>23</frames>
  </splice_time>
  <message>PTS[00:00:02.969]. Current NTP [16:21:23.573].</message>
  <splice_offset>0</splice_offset>
  <value>cue_point</value>
</response>
```

RETURNING FROM AN AVAIL

NAME	TYPE	RANGE	DESCRIPTION
return_offset	integer		The number of milliseconds to wait (also known as “preroll”) before returning.

Example:

```
<cue_point>
    <return_offset>0</return_offset>
</cue_point>
```

CANCELING AN AVAIL

NAME	TYPE	RANGE	DESCRIPTION
cancel_event_id	integer		The event ID of an upcoming avail to cancel.

Example:

```
<cue_point>
    <cancel_event_id>4</cancel_event_id>
</cue_point>
```

CUE POINT COMMAND RESPONSE

NAME	TYPE	RANGE	DESCRIPTION
event_id	integer		The event ID of the new event (either what you passed in, or auto-generated).
splice_time/hours	integer		The hour when the avail starts.
splice_time/minutes	integer		The minute when the avail starts.
splice_time/seconds	integer		The second when the avail starts.
splice_time/frames	integer		The frame (within hour+minute+second) when the avail starts.
splice_offset	integer		The offset from now (preroll, in milliseconds) at which this event occurs.
message	string		A more detailed human-readable log message.
errors/error/code	integer		Error code returned by this command.
errors/error/message	string		A more detailed human-readable error message.

Examples of success and failure:

```
<response value="cue_point">
    <event_id>8</event_id>
    <splice_time>
        <hours>0</hours>
        <minutes>0</minutes>
        <seconds>0</seconds>
```

```
        <frames>0</frames>
    </splice_time>
    <splice_offset>8000</splice_offset>
    <message>
        Inserted at PTS[1234]. Avail time[00:00:05.000] PTS[2345], duration[30].
    </message>
</response>

<response value="cue_point">
    <event_id>9</event_id>
    <errors>
        <error>
            <code>1040</code>
            <message>Preroll time must be positive integer</message>
        </error>
    <errors>
</response>
```

TIME SIGNAL PARAMETERS

INSERTING A TIME SIGNAL

NAME	TYPE	RANGE	DESCRIPTION
time/hours	integer		The hour (in local time) when the time signal occurs. Time signals scheduled later than 6 hours in the future are not supported.
time/minutes	integer		The minute (in local time) when the time signal occurs.
time/seconds	integer		The second (in local time) when the time signal occurs.
time/frames	integer		The frame (within hour+minute+second) when the time signal occurs.
offset	integer		The offset from now (preroll, in milliseconds) at which this event occurs.
descriptors	string		A hexadecimal string containing time signal descriptor data (optional).

TIME SIGNAL COMMAND RESPONSE

NAME	TYPE	RANGE	DESCRIPTION
signal_time/hours	integer		The hour (in local time) when the time signal occurs.
signal_time/minutes	integer		The minute (in local time) when the time signal occurs.
signal_time/seconds	integer		The second (in local time) when the time signal occurs.
message	string		A more detailed human-readable log message.
errors/error/code	integer		Error code returned by this command.
errors/error/message	string		A more detailed human-readable error message.

A splice_time (based on timecode_source) or an offset (in milliseconds) must be given. Example:

```
<time_signal>
    <time>
        <hours>0</hours>
        <minutes>0</minutes>
        <seconds>0</seconds>
        <frames>0</frames>
    </time>
    <descriptors>
        021B43554549000000027FBF030C54564E41313030303030303031300000
    </descriptors>
</time_signal>

<time_signal>
    <offset>1000</offset>
    <descriptors>
        021B43554549000000027FBF030C54564E41313030303030303031300000
    </descriptors>
</time_signal>
```

A detailed view of the descriptor data provided in the example:

```
descriptor_loop_length will be set to 29 (number of hex bytes)
scte35_descriptor_tag: 2
descriptor_length: 27
identifier: <binary data>
segmentation_event_id: 2
segmentation_event_cancel: 0
reserved: 127
program_segmentation_flag: 1
segmentation_duration_flag: 0
reserved: 63
segmentation_upid_type: 3 (Ad-ID)
```

```
segmentation_upid_length: 12
segmentation_upid: <binary data>
segmentation_type_id: 48 (Provider Advertisement Start)
segment_num: 0
segments_expected: 0
```

Examples of success and failure:

```
<response value="time_signal">
    <message>
        Inserted time signal at event time[1234], PTS[1234].
        Signal time[1234] PTS[1234].
    </message>
    <tag>1</tag>
    <signal_time>
        <hours>0</hours>
        <minutes>0</minutes>
        <seconds>0</seconds>
    </signal_time>
</response>

<response value="time_signal">
    <tag>1</tag>
    <errors>
        <error>
            <code>1040</code>
            <message>Invalid time signal message</message>
        </error>
    <errors>
</response>

<response>
  <tag>7</tag>
  <errors>
    <error>
      <code>1040</code>
      <message>AddTimeSignal failed. Preroll should be less than 13 hours 15 minutes.</message>
    </error>
  </errors>
  <value>time_signal</value>
</response>
```

METADATA INSERTION PARAMETERS

INSERTING TIMED METADATA

NAME	TYPE	RANGE	DESCRIPTION
time/hours	integer		The hour (in local time) to insert metadata. If <time>…</time> is omitted, insertion is immediate.
time/minutes	integer		The minute (in local time) to insert metadata.
time/seconds	integer		The second (in local time) to insert metadata.
time/frames	integer		The frame to insert metadata.
id3	Binary ID3 data		The ID3 data to insert into the stream. In general, each set of ID3 data must be preceded by the ID3 frame descriptor, such as ID3TIT2 for Title information or ID3TCOP for Copyright information. Available ID3 frames descriptors are available in the ID3 specification. The resulting string should then be provided in base64 encoding, specified by <id3 encoding="base64">(base64 encoded content)</id3>.
cancel	boolean		Cancels all pending insertion commands.

INSERTING PRIVATE METADATA

NAME	TYPE	RANGE	DESCRIPTION
time/hours	integer		The hour (in local time) to insert metadata. If <time>…</time> is omitted, insertion is immediate.
time/minutes	integer		The minute (in local time) to insert metadata.
time/seconds	integer		The second (in local time) to insert metadata.
time/frames	integer		The frame to insert metadata.
name	string		Name for this metadata entry.
parameters	list of parameters		List of <parameter> elements making up this metadata entry.
parameter/name	string		Name of parameter.
parameter/type	enum	string, number	Type of parameter.
parameter/value	string		Value of parameter. This can be provided in base64 encoding, specified by <value encoding="base64">(base64 encoded content)</value>.
cancel	boolean		Cancels all pending private metadata insertion commands.

Example command for inserting Private Metadata:

```
curl -H "content-type: application/xml" -H "accept: application/xml" -d @private_metadata_example.xml -X
```

Corresponding XML request body:

```
<?xml version="1.0" encoding="UTF-8"?>
<private_metadata>
    <name>first ad</name>
    <parameters>
        <parameter>
            <name>begin</name>
            <type>String</type>
            <value>30</value>
        </parameter>
        <parameter>
            <name>commercial_flag</name>
            <type>Number</type>
            <value>1</value>
        </parameter>
    </parameters>
</private_metadata>
```

AVAIL IMAGE PARAMETERS

AVAILIMAGEPARAMETERS

NAME	TYPE	RANGE	DESCRIPTION
uri	string		The local path or remote URL of the avail image
username	string		The username for URL/S3 authentication
password	string		The password for URL/S3 authentication

An avail_image must be given. Example:

```
<avail_image>
    <uri>http://example.com/protected/blanking1.png</uri>
    <username>brian</username>
    <password>secret123</password>
</avail_image>
```

Examples of success and failure:

```
<response value="avail_image">
  <tag>3</tag>
  <message>Cached and set avail blanking image to [http://example.com/protected/blanking1.png]</message>
  <value>avail_image</value>
</response>
```

```
<response value="avail_image">
  <tag>1</tag>
  <errors>
    <error>
      <code>5002</code>
      <message>Unable to fetch [http://example.com/protected/blanking1.png] during local caching.</messa
    </error>
  </errors>
  <value>avail_image</value>
</response>
```

BLACKOUT IMAGE PARAMETERS

BLACKOUTIMAGEPARAMETERS

NAME	TYPE	RANGE	DESCRIPTION
uri	string		The local path or remote URL of the blackout image

A blackout_image must be given. Example:

```
<blackout_image>
    <uri>/opt/elemental_se/web/public/example_files/Elemental_logo.png</uri>
</blackout_image>
```

Examples of success and failure:

```
<response value="blackout_image">
  <tag>3</tag>
  <message>Set blackout image to [/opt/elemental_se/web/public/example_files/Elemental_logo.png]</messag
  <value>blackout_image</value>
</response>

<response value="blackout_image">
  <tag>1</tag>
  <errors>
    <error>
      <code>1040</code>
      <message>Blackout image insertion is not enabled for this event.</message>
    </error>
  </errors>
  <value>blackout_image</value>
</response>
```

DEVICE

NAME	TYPE	RANGE	DESCRIPTION
name	string		A name to identify the device.

ROUTER

NAME	TYPE	RANGE	DESCRIPTION
name	string	string	A name to identify the router.
ip	string	Valid IP address	Specifies the IP address of the router.
max_inputs	integer	>= 1	Specifies the total number of inputs for the router.
router_input_device	Router Input Device		Specifies a particular router input device. A router has many router input devices (equal to the max_inputs).
max_outputs	integer	>= 1	Specifies the total number of outputs for the router.
router_output	Router Output		Specifies a particular router output. A router may have many router outputs (up to the max_outputs).
router_type	string	blackmagic_videohub, harris_panacea, miranda_nvision	Designates the router type.
*router*_settings	Router Settings	harris_panacea_settings‘, miranda_nvision_settings’	Router settings required by the specified router type. Note: replace *router* with the router type you are using in the XML tag.

ROUTER INPUT

NAME	TYPE	RANGE	DESCRIPTION
router_ip	string	Valid IP address	IP address of the router.
router_type	string	blackmagic_videohub, harris_panacea, miranda_nvision, snell_aurora, imagine_xypassthrough, imagine_lrc	Designates the router type.
*router*_settings	Router Settings	harris_panacea_settings‘, miranda_nvision_settings’	Router settings required by the specified router type. Note: replace *router* with the router type you are using in the XML tag.
input_number	integer	>= 1	Desired SDI input from the router.

ROUTER OUTPUT

NAME	TYPE	RANGE	DESCRIPTION
output_number	integer	>= 1	Output number of the router that is connected to the device.
device_id	integer		Device ID of HD-SDI input card that is connected to the router.

MPTS

NAME	TYPE	RANGE	DESCRIPTION
name	string		A name for the MPTS
bitrate	integer	>= 1	The total bitrate of the MPTS in bits/second
video_allocation	integer	>= 1	The bitrate allocated for video traffic in bits/second
transport_stream_id	integer	0 â€“ 65535	The value of the transport stream ID field in the Program Map Table.
buffer_msec	integer	0 – 10000	UDP output buffering in milliseconds. Larger values increase latency through the transcoder but simultaneously assist the transcoder in maintaining a constant, low-jitter UDP/RTP output while accommodating clock recovery, input switching, input disruptions, picture reordering, etc.
output_listening	boolean	true or **false**	When checked the specified multicast destination addresses are monitored (joined) and received packets are examined. If these destinations are served by other senders, then the UDP transmission from this event to that address is suppressed. If no packets from the other senders are detected for a period of output_listening_interval then the transmission from this event is no longer suppressed. When traffic from another sender is re-detected, the transmission from this event is immediately suppressed again. This feature supports very fast acting 1:1 failover scenarios.
output_listening_interval	integer		The specified detection interval for Output Listening feature, in milliseconds. If no packets are detected in the interval, local transmission is activated. Detection interval only specifies activation time. Deactivation time is immediate.
pat_interval	integer	10 â€“ 1000 (Default: **100**)	The PAT interval in ms
destination	Location		The primary destination for the MPTS output. Can be a UDP or RTP location.
secondary_destination	Location		The secondary destination for the MPTS output. Can be a UDP or RTP location.
fec_output_settings	FEC Output Settings		FEC settings for the MPTS output
mpts_members	Array of MPTS Members		An array of MPTS member channels

LOCAL MPTS MEMBER

NAME	TYPE	RANGE	DESCRIPTION
type	string	local	Specifies the type of MPTS member. To create a Local MPTS member, use “local”.
program_number	integer	1 â€“Â 65535	The program number to use for this MPTS member
pid_map	Hash		PID assignments to use for this member in the MPTS output. If no values are provided, PIDs will be assigned automatically. Supports both PID keys with single values (pmt_pid, video_pid, pcr_pid, scte35_pid, klv_data_pids, dvb_teletext_pid, etv_platform_pid, etv_signal_pid, timed_metadata_pid, private_metadata_pid) and those with multiple values (audio_pids, dvb_sub_pids, scte27_pids). See below for format example.
live_event_id	integer	The ID of any MPTS-eligible LiveEvent	The ID of the LiveEvent represented by this MPTS member channel

REMOTE MPTS MEMBER

NAME	TYPE	RANGE	DESCRIPTION
type	string	remote	Specifies the type of MPTS member. To create a Remote MPTS member, use “remote”.
program_number	integer	1 â€“Â 65535	The program number to use for this MPTS member
pid_map	Hash		PID assignments to use for this member in the MPTS output. If no values are provided, PIDs will be assigned automatically. Supports both PID keys with single values (pmt_pid, video_pid, pcr_pid, scte35_pid, klv_data_pids, dvb_teletext_pid, etv_platform_pid, etv_signal_pid, timed_metadata_pid, private_metadata_pid) and those with multiple values (audio_pids, dvb_sub_pids, scte27_pids). See below for format example.
input	Location		The primary input for this channel’s video content. Can be a UDP or RTP location; must be RTP if a secondary input is also provided.
secondary_input	Location		The secondary input for this channel’s video content. Must be an RTP location.
complexity_receipt_destination	Location		Primary multicast destination for complexity estimates from encoder to MPTS.
secondary_complexity_receipt_destination	Location		Additional multicast destination for complexity estimates from encoder to statmux. Typical use case is to enable multiple network path redundancy (two interfaces and two switches) from encoder to MPTS.
allocation_transmit_destination	Location		Multicast destination for bitrate allocations from MPTS to encoder.
secondary_allocation_transmit_destination	Location		Additional multicast destination for bitrate allocations from MPTS to encoder. Typical use case is to enable multiple network path redundancy (two interfaces and two switches) from encoder to MPTS.

PID MAP EXAMPLE

```
<pid_map>
  <pmt_pid>888</pmt_pid>
  <audio_pids>
    <audio_pid>240</audio_pid>
    <audio_pid>241</audio_pid>
  </audio_pids>
</pid_map>
```

SEQUENCER CONFIG

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
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
exclude_gpu_3	boolean		Prevents jobs from running on GPU 3
rtmp_input	boolean		Enables RTMP input
rtmp_input_port	integer		Sets the input RTMP input port number
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.
ingest_parser_enabled	boolean		If checked, monitors and parses SDI ingest information and issues alerts when necessary.

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 Live Event will error once it is started.

IDENTIFIER	FORMAT	DESCRIPTION
dt	YYYYMMDDTHHMMSS	UTC datetime of the start time of the Live Event. NOTE: HLS outputs will use the current datetime for each segment.
d	YYYYMMDD	UTC date of the start time of the Live Event. NOTE: HLS outputs will use the current date for each segment.
t	HHMMSS	UTC time of the start time of the Live Event. 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

IDENTIFIER	FORMAT	DESCRIPTION
job	Job ID	ID of the server job. Used to ensure unique output destinations.
$event$	Event ID	ID of the live event. Used to ensure unique output destinations.
$$	$	Escaped $

Fields that accept format identifier fields include:

- Destination URI (Archive, Apple HLS and DASH ISO Output Groups) (Timestamp and Filename identifiers only)
- Adobe RTMP Endpoint URI (Timestamp and Filename identifiers only)
- MS Smooth Publish Point URI (Timestamp and Filename identifiers only)
- Name Modifier (Archive, Apple HLS, MS Smooth and DASH ISO outputs)
- Adobe RTMP Stream Name

Name Modifiers for DASH ISO outputs accept three format identifiers: $Number$, $Time$, and $Bandwidth$. In the manifest, $Number$ allows SegmentTemplate to contain "duration" and "startNumber". Alternately, $Time$ gives the manifest a SegmentTimeline.

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

INPUT AND DESIRED OUTPUT		HOW TO GET THERE			
INPUT	OUTPUT	PREPROCESSOR FIELD	FORCE MODE FIELD (FORCE)	INTERLACE MODE FIELD	TELECINE MODE FIELD
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 Elemental Live 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 Live Event errors.

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

SNMP BASICS

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

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

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

returns

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

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

- https://<server_ip>/mib/ELEMENTAL_MIB.txt - Base MIB for all Elemental products
- https://<server_ip>/mib/ELEMENTAL_LIVE_MIB.txt - Objects specific to Elemental Live 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
```

SNMP OPERATIONS

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

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

VARIABLE	TYPE	GET VALUES	SET VALUES
ELEMENTAL-MIB::networkSettings	Integer	Will always return 1. Required for some network management systems	1 will load new network settings. Network settings are configured in the Elemental web interface.
ELEMENTAL-MIB::mountPoints	Integer	Number of user-mounted filesystems in /mnt	1 will load new mount settings. Filesystem mount settings are configured in the Elemental web interface.
ELEMENTAL-MIB::version	String	Product version	
ELEMENTAL-MIB::httpdStatus	Integer	0 if the httpd service is not running, 1 if the service is running	0 stops the httpd service. 1 starts the service, and 2 restarts the service
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 LIVE EVENT OPERATIONS

Live Events are controlled using the liveEventsTable from the ELEMENTAL-LIVE-MIB. The liveEventsTable provides the following variables:

VARIABLE	TYPE	GET VALUES	SET VALUES
ELEMENTAL-LIVE-MIB::eventId	Integer	Live Event ID (Used as the index to the liveEventsTable)	
ELEMENTAL-LIVE-MIB::eventName	String	Name of Live Event	
ELEMENTAL-LIVE-MIB::eventRunning	Integer	1 if the Live Event is currently running, 0 otherwise	0 will stop a Live Event that is currently running, 1 will start a Live Event that is not currently running and not in an archived state.
ELEMENTAL-LIVE-MIB::eventError	Integer	1 if the Live Event is in the error state, 0 otherwise	
ELEMENTAL-LIVE-MIB::eventStatus	String	Status information about the Live Event, in XML format	
ELEMENTAL-LIVE-MIB::nodeId	String	The numerical ID of the node the job is running on	

The user should use the web interface or REST interface to create Live Events in advance with the desired parameters. The Live Event can be started via SNMP as follows (eventId is 2):

```
snmpset -c elemental_snmp_write -v2c -m ELEMENTAL-MIB:ELEMENTAL-LIVE-MIB \
    localhost eventRunning.2 i 1
```

SNMP TRAPS

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

SNMP Traps are generated for the following events:

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

AUTHENTICATION

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

CONFIGURING AUTHENTICATION

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

```
cd /opt/elemental_se
```

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

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

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

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

MANAGING ROLES

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

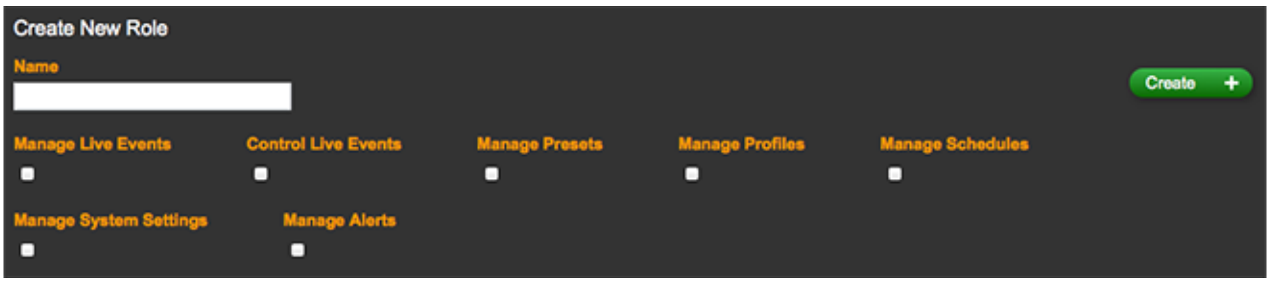
The Elemental Live system comes with a set of predefined Roles:

- **Admin**: The Admin role has access to the entire Elemental Live system
- **Manager**: The Manager role can create and edit Live Events, Presets and Profiles, and can control Live Events
- **Operator**: The Operator role can only control Live Events (Start, Stop, Reset, Archive, etc.)
- **Viewer**: The Viewer role has read-only access to the Elemental Live 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 Live Events**: Allows user to create and edit Live Events
- **Control Live Events**: Allows user to control the state of Live Events (Start, Stop, Reset, 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 Schedules**: Allows user to create and edit Schedules
- **Manage System Settings**: Allows user to update the Elemental Live 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.

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.
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 Elemental Audio Decode Package license option.
Dolby® E frames carried in PCM audio streams	Input	Decoding of Dolby E in PCM stream requires the 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 Elemental Advanced Audio Package license option.
Dolby® Digital Plus audio codec	Output	Encoding with Dolby Digital Plus™ requires the 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 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
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 DNxHD DV/DVCPRO DV25 DV50 DVCPro HD H.264 JPEG 2000 MPEG-2 Panasonic P2 SonyXDCam, SonyXDCam MPEG-4 Proxy	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 REAL-TIME INPUTS

	MEDIA TYPE	VIDEO CODECS	AUDIO CODECS
SDI		Uncompressed	Dolby® Digital PCM Dolby® Digital Plus™ Dolby® E frames carried in PCM streams
UDP/RTP	MPEG TS	H.264 HEVC (H.265) MPEG-2	AAC Dolby® Digital Dolby® Digital Plus™ Dolby® E frames carried in PCM streams MPEG Audio PCM
ASI	MPEG TS	H.264 HEVC (H.265) MPEG-2	AAC Dolby® Digital Dolby® Digital Plus™ Dolby® E frames carried in PCM streams MPEG Audio PCM
HTTP	HLS	H.264 HEVC (H.265)	AAC
RTMP		H.264	AAC

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

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

CONTAINERS AND CODECS FOR REAL-TIME OUTPUTS

	VIDEO CODECS	AUDIO CODECS
Apple® HTTP Live Streaming	H.264 HEVC (H.265)	AAC Dolby® Digital Dolby® Digital Plus™
DASH-ISO	H.264 HEVC (H.265)	AAC Dolby® Digital Dolby® Digital Plus™
ISMV for MSS	H.264	AAC Dolby® Digital Dolby® Digital Plus™
RTMP	H.264	AAC
UDP	H.264 HEVC (H.265) MPEG2	AAC Dolby® Digital Dolby® Digital Plus™

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 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 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 Live 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
Sample-based encryption with Key format and Key format versions attributes	HLS output group > Advanced > Alternate Manifest Destination > Encryption = value other than 1Mainstream	5
	HLS output group > Advanced > Alternate Manifest Destination > Encryption = 1Mainstream	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 on captions, including information on pass-through, on stripping out captions and on setting up for captions, see "Working with Captions - Quick Guide", available on the Elemental Technologies Knowledge Base.

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
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
SDI Stream Input	ARIB	
	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
	Teletext	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
UDP/RTP Stream Input	DVB-Sub	Burn-in

	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
	SCTE-27	Burn-in DVB-Sub
	Teletext	Burn-in 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
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
SDI Stream 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
SDI Stream Input	Teletext	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
UDP/RTP Stream Input	DVB-Sub	Burn-in
	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
	SCTE-27	Burn-in DVB-Sub
	Teletext	Burn-in 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
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
SDI Stream Input	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	Teletext	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
UDP/RTP Stream Input	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	DVB-Sub	Burn-in
	SCTE-27	Burn-in DVB-Sub
	Teletext	Burn-in

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
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
SDI Stream 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
	Teletext	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
UDP/RTP Stream Input	DVB-Sub	Burn-in
	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
	SCTE-27	Burn-in DVB-Sub
	Teletext	Burn-in

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
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
SDI Stream 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
	ARIB	
	Teletext	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
UDP/RTP Stream 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

RTMP 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 RTMP CaptionInfo RTMP CuePoint SCTE-20+Embedded
	SCTE-20	Burn-in Embedded Embedded+SCTE-20 RTMP CaptionInfo RTMP CuePoint SCTE-20+Embedded
MP4 Container in File Input	Embedded	Burn-in Embedded Embedded+SCTE-20 RTMP CaptionInfo RTMP CuePoint SCTE-20+Embedded
	SCTE-20	Burn-in Embedded Embedded+SCTE-20 RTMP CaptionInfo RTMP CuePoint SCTE-20+Embedded
MPEG Transport Streams in File Input	DVB-Sub	Burn-in
	SCTE-27	Burn-in DVB-Sub
	Teletext	Burn-in RTMP CuePoint
MXF Container in File Input	Ancillary Data	Burn-in Embedded Embedded+SCTE-20 RTMP CaptionInfo RTMP CuePoint SCTE-20+Embedded
	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 RTMP CaptionInfo RTMP CuePoint SCTE-20+Embedded

	Embedded	Burn-in Embedded Embedded+SCTE-20 RTMP CaptionInfo RTMP CuePoint SCTE-20+Embedded
	SCTE-20	Burn-in Embedded Embedded+SCTE-20 RTMP CaptionInfo RTMP CuePoint SCTE-20+Embedded
Raw (No Container) in File Input	SCC	Burn-in Embedded Embedded+SCTE-20 RTMP CaptionInfo RTMP CuePoint SCTE-20+Embedded
	SMI	Burn-in RTMP CuePoint
	SMTPE-TT	Burn-in RTMP CuePoint
	SRT	Burn-in RTMP CuePoint
	STL	Burn-in RTMP CuePoint
	TTML	Burn-in RTMP CuePoint
SDI Stream Input	ARIB	
	Embedded	Burn-in Embedded Embedded+SCTE-20 RTMP CaptionInfo RTMP CuePoint SCTE-20+Embedded
	SCTE-20	Burn-in Embedded Embedded+SCTE-20 RTMP CaptionInfo RTMP CuePoint SCTE-20+Embedded
	Teletext	Burn-in RTMP CuePoint
Transport Stream in File Input	Embedded	Burn-in Embedded Embedded+SCTE-20 RTMP CaptionInfo RTMP CuePoint SCTE-20+Embedded
	SCTE-20	Burn-in Embedded Embedded+SCTE-20 RTMP CaptionInfo RTMP CuePoint SCTE-20+Embedded
UDP/RTP Stream Input	DVB-Sub	Burn-in

	Embedded	Burn-in Embedded Embedded+SCTE-20 RTMP CaptionInfo RTMP CuePoint SCTE-20+Embedded
	SCTE-27	Burn-in DVB-Sub
	Teletext	Burn-in RTMP CuePoint
	SCTE-20	Burn-in Embedded Embedded+SCTE-20 RTMP CaptionInfo RTMP CuePoint SCTE-20+Embedded

UDP/TRANSPORT STREAM 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
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
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
SDI Stream Input	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	ARIB	ARIB
	Teletext	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
UDP/RTP Stream Input	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	DVB-Sub	Burn-in
	SCTE-27	Burn-in DVB-Sub
	Teletext	Burn-in

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
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
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
SDI Stream Input	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	ARIB	ARIB
	Teletext	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
UDP/RTP Stream Input	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	DVB-Sub	Burn-in
	SCTE-27	Burn-in DVB-Sub
	Teletext	Burn-in

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
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
SDI Stream 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
	Teletext	Burn-in CFF-TT
Transport Stream 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
UDP/RTP Stream Input	DVB-Sub	Burn-in

	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
	SCTE-27	Burn-in DVB-Sub
	Teletext	Burn-in CFF-TT

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
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
SDI Stream Input	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	Teletext	Burn-in Teletext
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
UDP/RTP Stream Input	Embedded	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	DVB-Sub	Burn-in DVB-Sub
	SCTE-20	Burn-in Embedded Embedded+SCTE-20 SCTE-20+Embedded
	SCTE-27	Burn-in
	Teletext	Burn-in Teletext