aws

User Guide

MediaLive

MediaLive: User Guide

Table of Contents

What Is AWS Elemental MediaLive?

AWS Elemental MediaLive is a real-time video service that lets you create live outputs for broadcast and streaming delivery.

You use MediaLive to transform live video content from one format and package into other formats and packages. You typically need to transform the content in order to provide a format and package that a playback device can handle. Playback devices include smartphones and set-top boxes attached to televisions.

Topics

AWS Elemental MediaLive Terminology

CDN

A content distribution network (CDN) is a network of servers that is downstream of the origin server or packager. The CDN distributes the content from the origin server to dozens or hundreds of networked servers that serve the content to your viewing users. This distributed network ensures that content can be delivered to thousands or millions of viewing users simultaneously.

Channel

A MediaLive channel ingests and transcodes (decodes and encodes) source content from the inputs that are attached to that channel, and packages the new content into outputs.

Channel class

Each channel belongs to one of the following classes:

- Standard class – a channel has two processing pipelines
- Single-pipeline class – a channel has one processing pipeline

Channel configuration

A MediaLive channel configuration contains information about how the channel ingests, transcodes, and packages content into output.

Downstream system

The *downstream system* is a set of one or more servers that is positioned after MediaLive in the workflow. The downstream system handles the content that is output from MediaLive.

Encode

An encode exists within an output. There are three types of encodes: video, audio, and captions. Each encode contains the instructions for one video stream, one audio stream, or one captions track that the transcoding process will create. Different encodes have different characteristics. For example, one video encode produced from the input might be high resolution while another is low resolution.

Input

A MediaLive input holds information that describes how the upstream system and the MediaLive channel are connected. The input identifies endpoints (IP addresses) in MediaLive (for a push input, where the upstream system pushes to MediaLive) or source IP addresses on the upstream system (for a pull input, where MediaLive pulls from the upstream system). MediaLive has different input types for different formats and protocols of the source content. For example, HLS input and RTMP Push input.

Input security group

A MediaLive input security group is a set of one or more ranges of IP addresses that define an allow list. You associate one or more input security groups with a push input in order to identify a range of IP addresses that are allowed to push content to the input.

Output

An output exists within an output group. It is a collection of encodes that you want to handle as one set.

Origin service

An origin service might be part of the downstream system that is positioned after MediaLive in the workflow. It accepts the video output from MediaLive.

Output Group

An output group is a collection of outputs within the MediaLive channel.

Packager

A packager might be part of the downstream system. It accepts the video output from MediaLive and repackages it. AWS Elemental MediaPackage is a packager.

Pipeline

In MediaLive, there are one or two separate and independent pipelines that perform the processing within the MediaLive input and the MediaLive channel.

Playback device

A playback device is the final component of the downstream system. It is the device that the people who are your audience use to view the video.

Schedule

Each MediaLive channel has an associated schedule. The schedule contains a list of actions to perform in the channel at a specific time.

Source content

The video content that MediaLive transcodes. The content typically consists of video, audio, captions, and metadata.

Upstream system

The system that is in front of MediaLive in the workflow and that holds the source content. Examples of an upstream system are a streaming camera or appliance that is directly connected to the internet, or a contribution encoder that is located in a stadium at a sports event.

Related Services

Amazon CloudWatch is a monitoring service for AWS Cloud resources and the applications that you run on AWS. Use CloudWatch to track MediaLive events about the progress of running channels and to view metrics about your resources.

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources for your users. Use IAM to control who can use your AWS resources (authentication) and what resources users can use in which ways (authorization).

AWS Elemental MediaPackage is a just-in-time video packaging and origination service that runs in the AWS Cloud. You can use AWS Elemental MediaPackage to package content that has been encoded by MediaLive.

AWS Elemental MediaConnect is a transport service for live video that runs in the AWS Cloud. You can use MediaConnect as a source for video to transcode.

AWS Elemental MediaStore is a video origination and storage service that offers the high performance and immediate consistency required for live and on-demand media. You can use AWS Elemental MediaStore to store assets that MediaLive retrieves and uses when transcoding, and as a destination for output from MediaLive.

AWS Resource Groups includes a tagging editor that lets you assign metadata to AWS resources. You can use Tag Editor to assign metadata to MediaLive channels and other resources.

Amazon Simple Storage Service (Amazon S3) is storage for the internet. You can use Amazon S3 to store assets that MediaLive retrieves and uses when transcoding, and as a destination for output from MediaLive.

AWS Systems Manager lets you store passwords in MediaLive in a secure manner, rather than storing them as plaintext. If you connect to external servers that you provide user credentials for, it is likely that you will have to use Systems Manager.

Amazon Virtual Private Cloud lets you set up your own virtual network within the AWS Cloud. Use Amazon VPC as the location for an upstream system, so that the transfer of source content is within a private cloud.

Accessing AWS Elemental MediaLive

You can access MediaLive using any of the following methods:

- **AWS Management Console** – The procedures throughout this guide explain how to use the AWS Management Console to perform tasks for AWS Elemental MediaLive.
- **AWS SDKs** – If you're using a programming language that AWS provides an SDK for, you can use an SDK to access AWS Elemental MediaLive. SDKs simplify authentication, integrate easily with your development environment, and provide easy access to MediaLive commands. For more information, see Tools for Amazon Web Services.
- **AWS Elemental MediaLive API** – If you're using a programming language that an SDK isn't available for, see the AWS Elemental MediaLive API Reference for information about API actions and about how to make API requests.
- **AWS Command Line Interface** – For more information, see the AWS Command Line Interface User Guide.

- **AWS Tools for Windows PowerShell** – For more information, see the AWS Tools for Windows PowerShell User Guide.

How AWS Elemental MediaLive works

From the point of view of AWS Elemental MediaLive, a live streaming workflow that includes MediaLive involves three systems:

- A MediaLive *channel*, which ingests and transcodes source content.
- One or more *upstream systems* that provide the *source content* (the video and other media) to MediaLive.

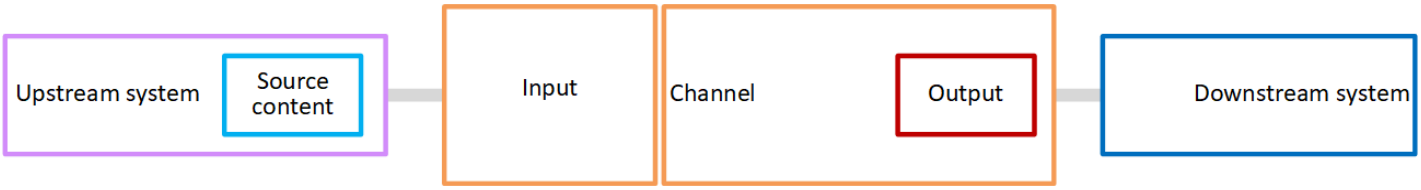
  Examples of an upstream system are a streaming camera or appliance that is directly connected to the internet, or a contribution encoder that is located in a sports stadium where a sports event is being held.

  The source content is in a specific package format and protocol. For example, the source content might be available as streaming HLS or streaming TS (transport stream). The source content contains video, audio, and optional captions streams that are in specific codecs or formats.
- One or more *downstream systems* that are the destinations for the output that MediaLive produces.

  A typical downstream system consists of an origin service or a packager that is connected to MediaLive, a content distribution network (CDN) that is downstream of the origin service or the packager, and a playback device or website where the users view the content. AWS Elemental MediaPackage is an example of an origin service and packager. Amazon CloudFront is an example of a CDN.

To create a MediaLive workflow, you create one or more MediaLive inputs. The inputs contain information about how MediaLive and the upstream system are connected. You also create a MediaLive channel and attach the inputs to the channel. The channel configuration data includes information about how MediaLive connects to the downstream systems.

This setup connects the components as illustrated in this diagram.



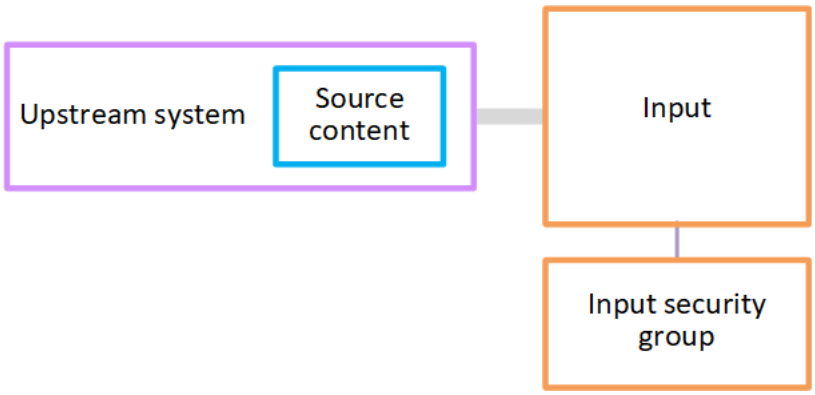
To start processing the content, you start the channel. When the channel is running, it ingests the source content from the upstream system that is identified by the input. The channel then transcodes that video (and the related audio, captions, and metadata) and creates outputs. MediaLive sends the outputs to the specified downstream systems.

Topics

AWS Elemental MediaLive inputs

An input contains information about how the upstream system and the channel connect to each other. The connection between the input and the upstream system might be a push (the upstream system pushes the content) or a pull (MediaLive pulls the content from the upstream system).

A push input has a MediaLive *input security group* associated with it. The input security group identifies a range of IP addresses that includes the source addresses on the upstream system. IP addresses within this range are allowed to push content to the input.



AWS Elemental MediaLive channels

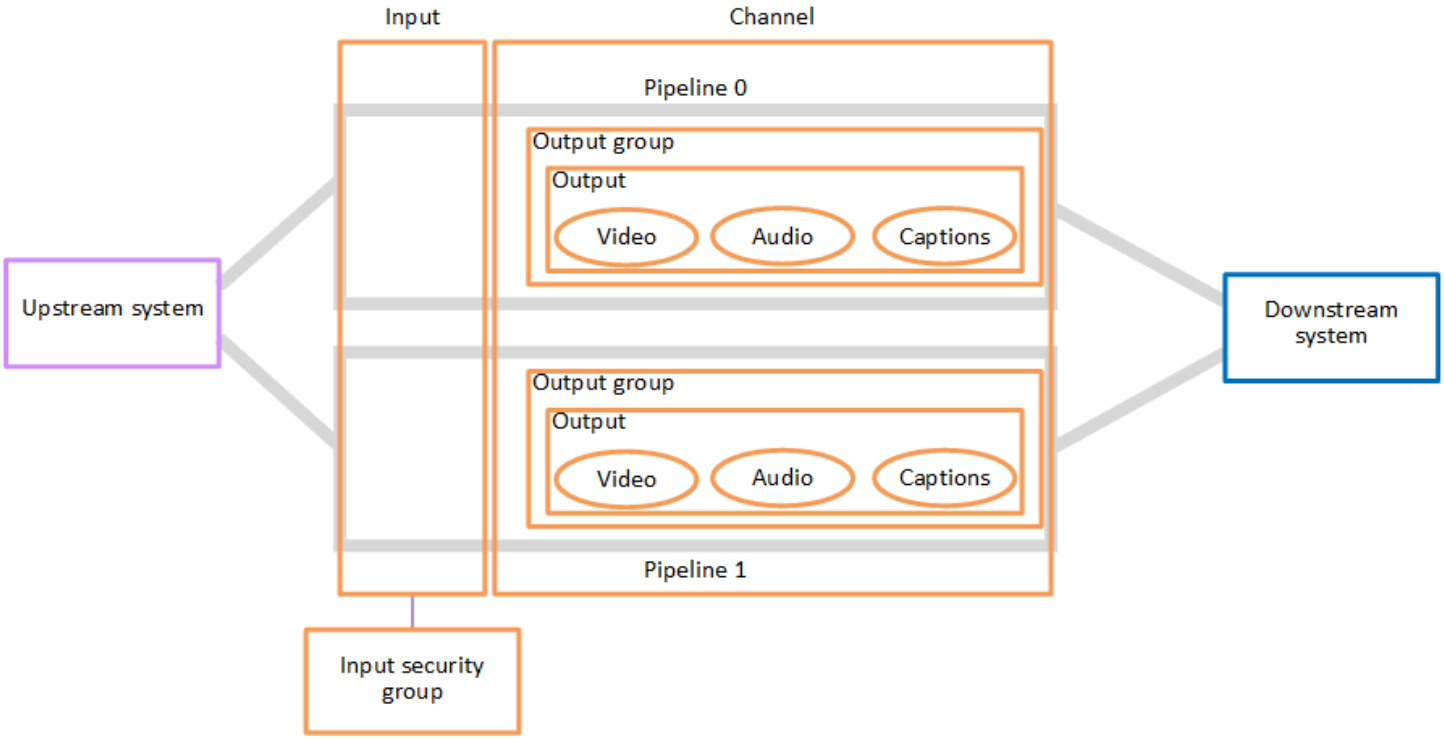
A channel can have several inputs attached to it, but it only ingests source content from one input at a time. (You use the channel *schedule* to set up the channel to switch from one input to another.)

The channel ingests the source content, transcodes it (decodes and encodes it), and packages it into *output groups*.

The channel contains one or more output groups. There are different types of output groups to handle the requirements of different downstream systems.

The output group consists of one or more *outputs*. Each output contains a specific combination of *encodes*. An encode is one video stream, one audio stream, or one captions track. Different encodes have different characteristics. The rules for combining encodes into outputs and for combining outputs into output groups depend on the type of the output group.

The following diagram is a detailed illustration of the workflow.



The illustration shows a channel with only one output group.

As another example, the channel might contain one HLS output group and one RTMP output group. The HLS output group might contain two outputs. One HLS output contains one high-resolution video, one audio, and one captions encode. The other HLS output contains one low-resolution video, one audio, and no captions. The RTMP output group contains one output that contains one video and one audio.

For information about designing this workflow and creating a channel, see *Setup: Preparing upstream and downstream* and *Setup: Planning the channel*.

AWS Elemental MediaLive pipelines

The processing within MediaLive occurs within one or two pipelines.

If you set up the workflow so that the channel and inputs have two pipelines (recommended), both pipelines work independently of each other but perform identical processing. Setting up with two pipelines provides resiliency within MediaLive.

With two pipelines, the upstream system must be set up to provide two sources, and the downstream system must be set up to receive two outputs.

AWS Elemental MediaLive schedule

Each MediaLive channel has one schedule associated with it. You add actions to the schedule to suit your requirements. There are different types of actions, including "switch input" (to switch to ingesting a different input) and "insert image overlay" (to overlay an image that you specify onto the video).

You can add these actions when the channel isn't running or when it is running. MediaLive sends the actions to the channel at the time identified in the schedule, and the channel performs the action.

For more information about schedules, see *Setup: Creating a schedule*

Pricing and reservations

This section contains information about two related topics: pricing for using AWS Elemental MediaLive, and creating reservations to obtain special pricing on your AWS Elemental MediaLive activities.

Topics

Pricing

As with other AWS products, there are no contracts or minimum commitments for using AWS Elemental MediaLive.

This section provides very general information about pricing. For detailed information, see https://aws.amazon.com/medialive/pricing/.

There are charges for MediaLive based on the state of resources. There are idle charges and running charges.

States

- A channel is either running or not running.

  It is *not running* if any of the following situations applies:

  - It has not been started
  - It was running but has failed and has not yet automatically restarted.
  - It was running but has been stopped for maintenance and has not yet automatically restarted.
- An input is idle or running.

  It is *idle* if either of the following situations applies:

  - It isn't attached to a channel
  - It is attached to a channel but the channel is not running.

Idle Charges

- There is an *idle channel charge* for each channel that isn't running. There isn't a channel charge for a running channel. The charges are for the inputs and outputs in the channel.
- There is an *idle push input charge* for each push input that isn't attached to a channel, and for each push input that is attached to a channel that isn't running.
- There is no charge for idle pull inputs.

Running Charges

- There is no channel charge for a running channel. There are charges for the inputs and outputs in the channel.
- There is a *running output charge* for each output that is configured in a channel that is running. The charge applies even if the output has been paused by the user or by Elemental Live.

  The charge for each output is based on the type of output, and on a combination of key video characteristics of the output, such as video output codec, and video frame rate. You specify the characteristics in the video settings of each output in the channel. For more information, see the section called “Set up video”.
- There is a *running input charge* for each input that is attached to a channel that is running. The charge applies to both push and pull inputs. It applies even to the inputs in the channel that aren't currently active or that aren't receiving any content.

  The input pricing is based on the type of input, and on a combination of key characteristics of the input, such as input codec, bitrate, and resolution. For detailed information about the basis for input pricing, see https://aws.amazon.com/medialive/pricing/. You specify some of these characteristics in the input specification when you create the channel. For more information, see the section called “Input specifications settings”.
- There is an *add-on charge* for running channels that have specific features enabled. The charge applies to the channel, not to individual inputs, outputs, or other components within the channel. For example, the add-on charge for Advanced Audio is applied at the same rate for a running channel with one output that uses advanced audio as it is for a running channel with three outputs that use advanced audio. For a list of add-on charges, see https://aws.amazon.com/medialive/pricing/.

Working with reservations in AWS Elemental MediaLive

A reservation is a one-year commitment to a particular input or output configuration. The reservation is allocated and billed on a monthly basis through that year. You can configure reservations to renew automatically.

You can purchase a reservation for processing resources that you are billed for, such as input processing, output processing, and add-ons like codec licenses. Each reservation gives you a reduced rate for the processing of the relevant resources.

You pay an hourly rate for a pool of minutes that your channels consume for one month. For information about charges for reservations, see the MediaLive price list.

> **Important**
>
> If some or all of the minutes in a reservation are not used in the month, those minutes are lost. The minutes are not transferred to the next month.

Topics

Input and output reservations

MediaLive offers input and output reservations.

Topics

Input reservation attributes and matching

An *input reservation* applies to the cost of processing input. An input reservation has these attributes:

- Codec
- Resolution (a range)
- Bitrate (a range)
- Region (in which the input runs)

How matching works

For a reservation to apply to an input, the attributes of the input reservation must match the fields in the channel's **Input specification**. The channel must also run in the Region that is specified in the reservation. For example, suppose that your input specification for a channel is **AVC**, **HD**, and **Max 20 Mbps**. A reservation that matches those attributes could apply to the input in that channel.

Output reservation attributes and matching

An *output reservation* applies to the cost of the processing output. An output reservation has the following attributes:

- Codec
- Resolution (a range)
- Bitrate (a range)
- Frame rate (a range)
- Region (in which the input runs)

How matching works

For a reservation to apply to an output, the attributes of the output reservation must match the corresponding fields in the channel configuration. The channel must also run in the Region that is specified in the reservation. You can find the fields on the AWS Elemental MediaLive console:

- For a regular video and audio output, the fields are in the **Video output** section of the channel configuration. To make most of the fields appear, you must choose a codec on the page.

- For an audio-only output, the fields are in the **Audio output** section of the channel configuration.

Example of matching

There is a match between an existing channel and a reservation if all of the fields in the channel match the corresponding reservation attributes.

There is a match if the value of a field in the channel is equal to, or falls within the range of, the corresponding attribute. For example, a frame rate of **`29.97 fps`** in the channel configuration falls within the range of a frame rate attribute of **`<=30fps`** in the reservation.

For the frame rate attribute, there is a match as follows:

- If the channel output frame rate is set to a specific frame rate: There is a match if the frame rate specified in the channel configuration falls within the reservation frame rate range. For example, the specified frame rate is **`24fps`** and the reservation is **`<=30fps`**.
- If the channel output frame rate is set to initialize from the source, there is a match only if the reservation range includes **`60fps`**. For example, there is a match on reservations with **`30-60fps`**.

  Note: If you purchase a reservation to target a specific output, and that output has the frame rate set to initialize from source, make sure that you purchase a reservation that specifies **`30-60fps`**. Don't purchase a reservation that specifies **`<=30fps`**.

Example of no match

If only one of the fields does not match its corresponding reservation attribute, there is no match between the output and reservation.

How an input or output reservation is applied

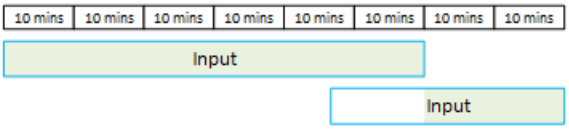
At the start of each monthly billing cycle, AWS replenishes each reservation with the pool of minutes for the month.

At the end of the cycle, AWS applies the minutes from a given reservation to reduce the cost for the processed items (inputs or outputs) whose attributes match this reservation. For each minute in the month, AWS determines if one or more matching items were running. It accumulates these *running minutes* within the hour, up to a maximum of 60 minutes in the hour.

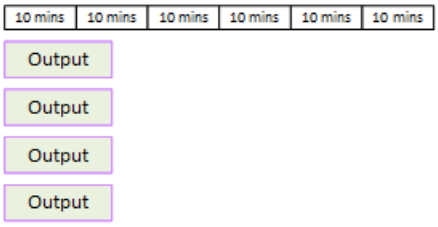
After the reservation minutes are used up for the hour, AWS charges the regular rate-per-minute for the remainder of the items in that hour.

Running minutes can be allocated over items

The running minutes could come from more than one item. For example, you start Channel A with an input that matches a given reservation. You have purchased only one instance of this reservation. After 45 minutes, you start Channel B that also has an input that matches a given reservation. After 15 more minutes, you stop Channel A. The running minutes are accumulated as shown by the shading in the following illustration.



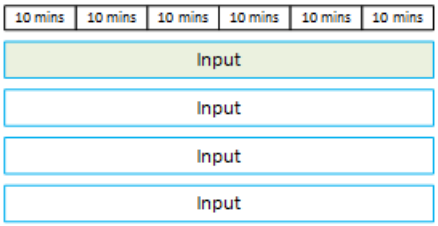
Here is another example of how different items can consume the running minutes. Suppose that in one hour, you run only outputs that match a given reservation. You have purchased only one instance of this reservation. You run these four matching outputs simultaneously for 15 minutes each. During that hour, you don't run any other matching outputs. Those four outputs would all contribute to the 60 minutes.



Processing bursts are not supported

The 60-minute rule means that reservations can't be used for processing bursts.

For example, in one hour you run four outputs that match a given reservation. You have purchased only one instance of this reservation. You run these four matching outputs simultaneously for 60 minutes each. Only one of these outputs is eligible for the reservation because one output is enough to use up the 60 running minutes per hour.



Unused minutes

If some or all of the minutes in the reservation are not used in the month, those minutes are lost.

The minutes are not transferred to the next month.

Running minutes can be allocated over items

There are no restrictions regarding channels:

- For example, the reservation could be consumed based on the processing of one input from one channel and another input from a different channel.
- There is no requirement that all the inputs or outputs in a given channel must be covered by a reservation.

Add-on reservations

Reservations are available for those items in the MediaLive price list that are considered to be add-ons, such as codec licenses.

An add-on reservation applies to the cost of the add-on for the entire channel. The reservation reduces the cost of the add-on regardless of how many times the add-on applies to the channel. For example, if three outputs in the same channel both use an advanced audio codec, you need only one reservation to reduce the cost of the add-on. You don't need three reservations for this channel.

Topics

Reservation attributes

The add-on reservations have these attributes:

- Add-on (Advanced Audio or Audio Normalization)
- Region (in which the channel is running)

How an add-on reservation is applied

At the start of each monthly billing cycle, AWS replenishes each add-on reservation with the pool of minutes for the month.

At the end of the cycle, AWS applies the minutes from a given reservation to reduce the cost for channels that use the add-on. For each minute in the month, it determines if one or more matching channels were running. A channel matches the reservation if the add-on feature is enabled.

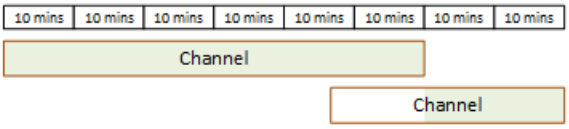
AWS accumulates these running minutes within the hour, up to a maximum of 60 minutes. After the reservation minutes are used up for the hour, AWS charges the regular rate-per-minute for the remainder of those channels for that hour.

Add-ons are per channel

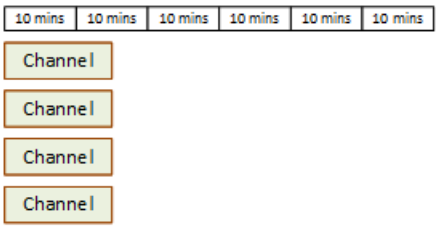
A channel matches the reservation if the add-on feature is enabled one or more times. Within one channel, the number of outputs that use the add-on isn't relevant. The reservation is consumed only once for the entire channel. For example, if there are two outputs in one channel that enable audio normalization, only one reservation is consumed.

Running minutes can be allocated over channels

The rule applies that applies to input and output reservations also applies to add-ons, except that the item is always a channel. For example, you start Channel A with two outputs that match the Advanced Audio reservation. You have purchased only one instance of this reservation. After 45 minutes, you start Channel B that has one output that matches the same reservation. After 15 more minutes, you stop Channel A. The running minutes are accumulated as shown by the shading in the following illustration.

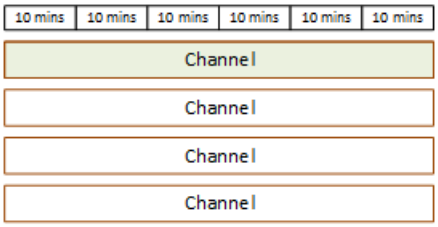


Here is another example of how different channels can consume the running minutes. Suppose that in one hour you run only channels that match the Advanced Audio reservation. You have purchased only one instance of this reservation. You run these four matching outputs simultaneously for 15 minutes each. During that hour, you don't run any other matching outputs. Those four outputs would all contribute to the 60 minutes.



Licensing bursts are not supported

The bursting rule that applies to input and output reservations also applies to add-on reservations, except that the item is always a channel. For example, in one hour you run four channels that match the Advanced Audio reservation. You have purchased only one instance of this reservation. You run these four matching channels simultaneously for 60 minutes each. Only one of these channels is eligible for the reservation because one channel is enough to use up the 60 running minutes per hour.



Unused minutes

At the end of the cycle, if some or all of the minutes in the add-on reservation are not used, those minutes are lost. Minutes are not transferred to the next month.

Purchasing a reservation

On the console, use the **Reservations** tab to purchase one or more reservations.

To purchase a reservation (console)

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Reservations**, and then choose **Reserve offerings**. On the **Offerings** page, complete the **Filter offerings** section to filter for specific offerings. For more information, see Filtering on the Offering Page.
3. Choose an offering.

If you are purchasing a reservation to target a specific output, and that output has the frame rate set to initialize from source, make sure that you purchase a reservation that specifies **`30-60fps`**. Don't purchase a reservation that specifies **`=30fps`**.

4. Select a number for the specific offering you are choosing, enter it in the **Count** field. For example, if you are reserving five HD AVC outputs, enter **5** in the **Count** field.
5. Choose **Add to cart**. The **Cart** tab title in the upper-left pane increments to show the total offerings currently in the cart. To remove an offering that you added to the cart, switch to the **Cart** tab.
6. To view the cart contents, choose the **Cart** tab. Optionally, while viewing the cart contents, you can set the reservation to automatically renew by selecting **Enable auto-renewal**. This option is off by default.
7. To purchase all of the offerings that are displayed on the **Cart** tab, choose **Purchase**.

> **Important**
>
> You can't cancel a reservation after you have purchased it.

Filtering on the offerings page

The **Offerings** page shows the different reservations that you can purchase.

Input and output offerings, which are described as follows:

- Resolution – Codec – Input/output – Bitrate – Frame rate (for outputs only) – Region

  For example: UHD AVC input at 10-20 Mbps in US West (Oregon)

Channel (add-ons) offerings, which are described as follows:

- Add-on – Region

  For example: Advanced Audio reserved outputs in US West (Oregon)

You can filter the offerings by using the filters in the left pane, as follows:

- Filter reservation type: input, output, or channel (for add-ons).

- Filter offerings based on attributes, such as resolution or bitrate.
- Use the **Match existing channel** filter to show only offerings that match the inputs and outputs in the chosen channel.
- Use the **Special feature** filter to show only add-on offerings.

Filtering does not affect the items in the cart.

Viewing purchased reservations

On the console, you can view the reservations that you have purchased. You can also view or edit any auto-renewals that you have created.

To view your purchased reservations (console)

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Reservations**.

   The information displayed for each reservation includes its expiry or auto-renewal date in the **Expiration** column.

To view your auto-renewals (console)

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Reservations**. Any existing auto-renewals are visible in the **Expiration** column.
3. To view more details, select a reservation by clicking on the **Name**.
4. In addition to other details, the auto-renewal status is displayed in the **Auto-renewal** section.

To edit your auto-renewals (console)

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Reservations**.
3. Select the reservation that you want to edit by clicking on the **Name**.
4. In the **Auto-renewal** section, select **Edit auto-renewal**.
5. In the **Edit reservation auto-renewal** window, you can turn auto-renewal on or off. You can also change the **Renewal count**.

6. Select **Save** to confirm the changes or **Cancel** to discard them.

Deleting an expired reservation

When a reservation has expired, you can delete the reservation from the list.

You can't cancel a reservation that hasn't expired.

To delete an expired reservation (console)

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Reservations**.
3. Choose the item or items, and then choose **Delete**.

Quotas in AWS Elemental MediaLive

There are quotas (formerly referred to as limits) that apply to the resources and operations of AWS Elemental MediaLive. A *quota* is a resource or operation cap that you can increase.

Requesting a quota increase

Use the Service Quotas console to request an increase on any quota and to view information about your current quotas.

Quotas versus constraints

MediaLive has quotas. It also has *constraints*, which are limits that you can't change. For more information about these constraints, see *Feature rules and limits*.

> **Note**
>
> There is a limit on the number of actions that a channel schedule can contain. This limit isn't listed here because it's not a quota that you can change. This limit is documented in *Feature rules and limits*.

AWS Elemental MediaLive feature rules and limits

The following table provides a summary of many of the rules and constraints that apply to AWS Elemental MediaLive features. You can't change any of these constraints.

MediaLive also includes quotas, which you can change. For more information about quotas, see *Quotas*.

Topics

Limits for inputs

Resource or feature	Constraint or rule
Input number, push inputs	You can attach 0 to 2 push inputs to a channel.
Input number, pull inputs	You can attach up to 20 inputs to a channel. After you have counted the push inputs, the remainder can be pull inputs.
Input number, CDI inputs	You can attach 0 or 1 regular CDI inputs to a channel. This input is a push input, so it counts towards the maximum number of push inputs in the channel. You can attach one set of partner CDI inputs to a channel. Attaching this set uses up the maximum number of push inputs in the channel. For information about these inputs, see the section called “CDI inputs as partner inputs”.

Resource or feature	Constraint or rule
Input number, Elemental Link inputs	You can attach up to 2 Elemental Link inputs to a channel. Elemental Link inputs are push inputs, so each counts towards your maximum number of push inputs in the channel. • You can attach these two Elemental Link inputs to one standard channel in order to implement pipeline redundancy. • You can include one or both of these Elemental Link inputs in a multiple-input channel, as part of an input switching workflow.
Input number, Elemental Link inputs per AWS Elemental Link hardware device	You can create up to 4 inputs (Link inputs) from each AWS Elemental Link hardware device. You can then attach each input to a different channel.
Input types – in automatic input failover	You can set up two push inputs as an automatic input failover pair. You can't set up pull inputs as a failover pair. The failover pair uses up your maximum number of push inputs.
Input types – for dynamic inputs	Only MP4 and Transport Stream (TS) file inputs that are stored in Amazon S3 or AWS Elemental MediaStore be set up as dynamic inputs.

Resource or feature	Constraint or rule
Input types – in multiple-input channels	You can attach multiple inputs to a channel, in order to implement input switching. You can't include an HLS input that is a VOD asset. For the definition of a VOD asset, see the section called “Support for live and file sources”.
	For the inputs that you attach in order to implement input switching, there are restricti ons related to input types and Availability Zones: • You can have multiple MediaConnect inputs attached to one channel, but all those inputs must be in the same two Availability Zones. • You can have multiple VPC inputs attached to one channel, but all these inputs must be in the same two Availability Zones. VPC inputs include CDI inputs, RTP VPC inputs, and RTMP VPC inputs. • If the channel has both MediaConnect inputs and VPC inputs, all these inputs must be in the same two Availability Zones.
Input – audio and captions selectors	Maximum of 32 audio and captions selectors (in any combination) in one channel.

Resource or feature	Constraint or rule
Input – captions selectors for OCR conversion	A maximum of three captions selectors that will use OCR conversion, per input. A selector uses OCR conversion if the specified format is DVB-Sub or SCTE-27, and at least one output encode that uses the selector is a WebVTT encode. If the selector is used in more than one WebVTT encode (for example, in two output groups), the selector counts only once towards the limit.
Input charges	Input from an AWS Elemental Link UHD device is charged at one rate. There are not separate rates for different resolutions in the content.

Limits for outputs

Resource or feature	Constraint or rule
Output, types	Maximum of one archive output groups in a channel. For information about output types, see the section called “Supported output types”.
Output encodes, frame capture	For frame capture encodes: • Maximum of three frame capture encodes in a channel. The single encode in a Frame Capture output group, and each (optional) frame capture encode in an HLS output group both count towards this limit. • Maximum of three frame capture outputs in each HLS output group.

Resource or feature	Constraint or rule
	For information about output types, see the section called “Supported output types”.
Output video encodes, UHD resolution, and input type	A channel with a CDI input allows one UHD output encode (maximum). The maximum number of *channels* with UHD is a quota that you can change, as described in *Quotas*. If you are using a CDI input, the maximum number of UHD *outputs* is a limitation. You can't change it.
Output video encodes, resolutions, and codecs	Standard definition (SD) video is supported with all codecs. For information about supported output codecs, see the section called “Supported codecs by output type”.
	High definition (HD) video is supported with H.264 and H.265.
	Ultra-high definition (UHD or 4K) video is supported with H.264 and H.265. For information about output video resolutio ns, see the section called “Supported codecs by output type”.
Output – audio encodes	Maximum of 33 audio encodes in one channel.

Limits for other features

Resource or feature	Constraint or rule
Color space, 3D LUT files in a channel	Maximum of 8 files in each channel.

Resource or feature	Constraint or rule
	For information about using 3D LUT files when converting color space, see the section called "3D LUT files".
Image Overlays	Maximum of eight different overlays (layers) active at one time in a channel. This means that the video can show up to eight different overlays at the same time. For information about image overlay, see the section called "Image overlays".
Motion graphic overlay	Maximum of one motion graphic overlay active at one time in a channel. For information about motion graphic overlay, see the section called "Motion graphics overlay".
Multiplexes	Each multiplex produces only one MPTS. For information about multiplex, see the section called "Multiplex and MPTS".
	All multiplex programs must include video.
Multiplexes, programs in a multiplex	Maximum of 20 programs per multiplex.
	Each program in a multiplex is single use. It is attached only to one multiplex, and you can use it only for that multiplex.
Multiplexes, channels in a multiplex	Each channel contains one and only one output group, of type multiplex. It can't contain any other type of output group.

Resource or feature	Constraint or rule
	Each channel is single use. You can attach it to only one program in the multiplex. You can use it only for that multiplex.
Output locking feature	Output locking is supported only with HLS and Microsoft Smooth. Although you enable the feature globally (for the entire channel), it only works with HLS output groups and Microsoft Smooth output groups.
Resiliency, automatic input failover	The automatic input failover applies to inputs, not to the entire channel. You can set up failover in only two, paired, inputs. The inputs must be push inputs.
Resiliency, pipeline redundancy	The pipeline redundancy feature (channel class) applies to the channel and all its inputs. The following rules apply to the channels and inputs: • Standard channel – You can attach only standard-class inputs. • Single-pipeline channel – You can attach single-class inputs (to omit support for pipeline redundancy) or standard-class inputs (to allow for easy upgrade of the channel at a later date).
Schedule, maximum number of actions	The schedule can contain a maximum of 1500 actions. You can't change this maximum. This maximum includes stale actions, actions that are in progress, and actions that aren't yet active. If you are near this maximum, you should delete stale actions.

Resource or feature	Constraint or rule
Schedule and input switches	The schedule can contain any number of scheduled input switching actions. For information about input switching, see the section called “Input switching”.
	You can switch to a specific input as many times as you want.

Limits for API requests

The following limits exist for API requests. For information about the current maximums (quotas) and about how to request an increase on any quota, see the Service Quotas console.

Type of request	Rule
Frequency of API requests, not including requests to the thumbnails API	Maximum 5 steady-state TPS (transactions per second). This limit is not a quota that you can increase.
	Maximum 30 burst. This limit is not a quota that you can increase.
Frequency of requests to the thumbnails API. For more information, see the section called “Limit on thumbnails”	There is a maximum to the TPS for thumbnail requests. This limit is a quota that you can increase. For the current quota, and to request an increase on the quota, see the Service Quotas console.

Getting started: IAM permissions for MediaLive

This section provides procedures to set up your organization to use AWS Elemental MediaLive. It also providers information about determining the IAM permissions that users and other AWS identities require. These permissions let you impose restricted controls on users and other AWS identities, in conformance with the security policies and procedures of your organization.

Topics

Preliminary steps for setting up

This topic describes preliminary steps, such as creating an account, to prepare you to use MediaLive. You aren't charged for these preliminary items. You are charged only for AWS services that you use.

Topics

Sign up for an AWS account

If you do not have an AWS account, complete the following steps to create one.

To sign up for an AWS account

1. Open https://portal.aws.amazon.com/billing/signup.
2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

When you sign up for an AWS account, an *AWS account root user* is created. The root user has access to all AWS services and resources in the account. As a security best practice, assign administrative access to a user, and use only the root user to perform tasks that require root user access.

AWS sends you a confirmation email after the sign-up process is complete. At any time, you can view your current account activity and manage your account by going to https://aws.amazon.com/ and choosing **My Account**.

Create a user with administrative access

After you sign up for an AWS account, secure your AWS account root user, enable AWS IAM Identity Center, and create an administrative user so that you don't use the root user for everyday tasks.

Secure your AWS account root user

1. Sign in to the AWS Management Console as the account owner by choosing **Root user** and entering your AWS account email address. On the next page, enter your password.

   For help signing in by using root user, see Signing in as the root user in the *AWS Sign-In User Guide*.

2. Turn on multi-factor authentication (MFA) for your root user.

   For instructions, see Enable a virtual MFA device for your AWS account root user (console) in the *IAM User Guide*.

Create a user with administrative access

1. Enable IAM Identity Center.

   For instructions, see Enabling AWS IAM Identity Center in the *AWS IAM Identity Center User Guide*.

2. In IAM Identity Center, grant administrative access to a user.

   For a tutorial about using the IAM Identity Center directory as your identity source, see Configure user access with the default IAM Identity Center directory in the *AWS IAM Identity Center User Guide*.

Sign in as the user with administrative access

- To sign in with your IAM Identity Center user, use the sign-in URL that was sent to your email address when you created the IAM Identity Center user.

  For help signing in using an IAM Identity Center user, see Signing in to the AWS access portal in the *AWS Sign-In User Guide*.

Assign access to additional users

1. In IAM Identity Center, create a permission set that follows the best practice of applying least-privilege permissions.

   For instructions, see Create a permission set in the *AWS IAM Identity Center User Guide*.
2. Assign users to a group, and then assign single sign-on access to the group.

   For instructions, see Add groups in the *AWS IAM Identity Center User Guide*.

Download tools

The AWS Management Console includes a console for AWS Elemental MediaLive, but if you want to access the services programmatically, see the following:

- The API guides document the operations that the services support and provide links to the related SDK and CLI documentation:
  - AWS Elemental MediaLive API Reference
- To call an API without having to handle low-level details like assembling raw HTTP requests, you can use an AWS SDK. The AWS SDKs provide functions and data types that encapsulate the functionality of AWS services. To download an AWS SDK and access installation instructions, see the applicable page:
  - Go
  - JavaScript
  - .NET
  - Node.js
  - Python
  - Ruby

For a complete list of AWS SDKs, see Tools for Amazon Web Services.

- You can use the AWS Command Line Interface (AWS CLI) to control multiple AWS services from the command line. You can also automate your commands using scripts. For more information, see AWS Command Line Interface.
- AWS Tools for Windows PowerShell supports these AWS services. For more information, see AWS Tools for PowerShell Cmdlet Reference.

Setting up IAM permissions for users

This section describes the permissions that you must assign to users and other AWS identities so that they can work with AWS Elemental MediaLive and other AWS services that your workflows use. After you have identified the required permissions, you will be able to design and create the relevant policies, and attach those policies to groups of users or to roles.

This section assumes that you have already performed these tasks:

- You have performed the initial setup described in the section called "Preliminary setup steps" in order to sign up for MediaLive and to create an administrator.
- You have read the recommendations in the section called "Identity and Access Management"about how to create administrators, users, and other AWS identities.

Topics

- Requirements for AWS Elemental Link
- Requirements for AWS Elemental MediaConnect
- Requirements for AWS Elemental MediaPackage
- Requirements for AWS Elemental MediaStore
- Requirements for AWS Resource Groups—tagging
- Requirements for Amazon S3
- Requirements for AWS Systems Manager—creating password parameters in parameter store

Reference: summary of non-administrator user access requirements

The following table shows all the types of permissions that you might need to assign to users. Each row in the column describes an activity or set of related activities that you might want to allow the user to perform. The last column lists the IAM actions that control access to those activities.

If this table doesn't provide enough information for you to determine which permissions to assign to users, see the alphabetical list of services that follow this section.

General activity that the user can perform	Corresponding service in IAM	Specific activities the user can perform	Actions to include in the policy

Use the features of MediaLive	MediaLive	Create, modify, and delete channels, devices, inputs, and input security groups	`CreateChannel`
			`CreateInput`
			`CreateInputSecurityGroup`
			`DeleteChannel`
			`DeleteInput`
			`DeleteInputSecurityGroup`
			`UpdateChannel`

General activity that the user can perform	Corresponding service in IAM	Specific activities the user can perform	Actions to include in the policy

			`UpdateInput`
			
			`UpdateInputDevice`
			
			`UpdateInputSecurityGroup`

	MediaLive	View channels, devices, inputs, and input security groups	`ListChannels`
			
			`ListInputDevices`
			
			`ListInputs`
			
			`ListInputSecurityGroups`
			
			`DescribeChannel`
			
			`DescribeInput`
			
			`DescribeInputDevice`
			
			`DescribeInputDeviceThumbnail`
			
			`DescribeInputSecurityGroup`

General activity that the user can perform	Corresponding service in IAM	Specific activities the user can perform	Actions to include in the policy

	MediaLive	Perform a batch operation on several channels or inputs or multiplexes or input security groups	`BatchDelete`
			`BatchStart`
			
			`BatchStop`

	MediaLive	Create or cancel an outgoing device transfer, or accept or reject an incoming device transfer, and view pending device transfers	`AcceptInputDeviceTransfer`
			
			`CancelInputDeviceTransfer`
			
			`ListInputDeviceTransfers`
			
			`RejectInputDeviceTransfer`
			
			`TransferInputDevice`

	MediaLive	Work with schedules	`DescribeSchedule`
			
			`BatchUpdateSchedule`

General activity that the user can perform	Corresponding service in IAM	Specific activities the user can perform	Actions to include in the policy

	MediaLive	Create or modify multiplexes	`CreateMultiplex`
			
			`DescribeMultiplex`
			
			`ListMultiplexes`
			
			`UpdateMultiplex`

	Amazon EC2		`DescribeAvailabilityZones` You need this operation to view the list of Availability Zones on the MediaLive console, so that you can choose two for the multiplex.

	MediaLive	Delete multiplexes	`DeleteMultiplex`
			
			`DescribeMultiplex`
			
			`ListMultiplexes`

	MediaLive	View multiplexes	`DescribeMultiplex`
			
			`ListMultiplexes`

	MediaLive	Change the class for a channel	`UpdateChannelClass`

General activity that the user can perform	Corresponding service in IAM	Specific activities the user can perform	Actions to include in the policy
	MediaLive	Run channels	`StartChannel` `StopChannel`
	MediaLive	Pause channels	Pause is an activity within the schedule feature, shown earlier in this table.
	MediaLive	Run multiplexes	`StartMultiplex` `StopMultiplex`

	MediaLive	Attach tags to channels, inputs, and input security groups when creating those resources	`CreateTag`
			`DeleteTags`
			`ListTagsForResources`

	MediaLive	Create, modify, delete, and view reservations and offerings	`DeleteReservation`
			`DescribeOffering`
			`DescribeReservation`
			`ListOfferings`
			`ListReservations`
			`PurchaseOffering`

General activity that the user can perform	Corresponding service in IAM	Specific activities the user can perform	Actions to include in the policy

	AWS CloudFormation	Create and delete the AWS CloudFormation stack. These permissions are always required. For example, if a user is using the workflow wizard and doesn't have `CreateStack` access, MediaLive will fail to create the workflow.	`ListStacks`
			
			`DescribeStacks`
			
			`DescribeStackResources`
			
			`CreateStack`
			
			`DeleteStack`

	CloudFront	Create and delete a CloudFront distribution, if your organization supports MediaPackage as an output destination.
		
		Note how the required permissions here are very different from the permissions because the workflow wizard actually creates the distribution.
			`ListDistributions`
			
			`DescribeDistribution`
			
			`CreateDistribution`
			
			`DeleteDistribution`

General activity that the user can perform	Corresponding service in IAM	Specific activities the user can perform	Actions to include in the policy

	Amazon EC2	Create a VPC input – View the VPC subnets and VPC security groups on the MediaLive console	`DescribeSubnets`
			
			`DescribeSecurityGroups`

	Amazon EC2	Set up a channel for delivery of output via your VPC – View the VPC subnets and VPC security groups on the MediaLive console.	`DescribeSubnets`
			
			`DescribeSecurityGroups`

	Amazon EC2	Set up a channel for delivery of output via your VPC – View the Elastic IP addresses on the console. The console finds the Elastic IP addresses that have been allocated for use in your AWS account.	`DescribeAddresses`

General activity that the user can perform	Corresponding service in IAM	Specific activities the user can perform	Actions to include in the policy

	MediaConnect	Use the workflow wizard to create a MediaConnect flow, if your organization supports sources from MediaConnect.
		
		Use the workflow wizard to delete a workflow that includes a source from MediaConnect.
			`List*`
			
			`Describe*`
			
			`Create*`
			
			`Delete*`

	MediaPackage	On the MediaLive console, view the MediaPackage channels in the dropdown list on the MediaLive channel.	`Describe*`

		Use the workflow wizard to create a MediaPackage channel, if your organization supports MediaPackage as an output destination.
		
		Use the workflow wizard to delete a workflow that includes a MediaPack age output.
			`List*`
			
			`Describe*`
			
			`Create*`
			
			`Delete*`

General activity that the user can perform	Corresponding service in IAM	Specific activities the user can perform	Actions to include in the policy

	MediaStore	Use the workflow wizard to create a MediaStore container , if your organization supports MediaStore as an output destination.
		
		Use the workflow wizard to delete a workflow that includes a MediaStore output.
			`List*`
			
			`Describe*`
			
			`Create*`
			
			`Delete*`

Monitor channel health	CloudWatch		`ListMetrics`
			
			`GetMetricData`
			
			`GetMetric Statistics`

Set up events	CloudWatch Events		All actions The managed policy `CloudWatchEventsFullAccess` provides these permissions

Set up channel logging	Amazon CloudWatch Logs	View logs	`FilterLogEvents`
			
			`GetLogEvents`

General activity that the user can perform	Corresponding service in IAM	Specific activities the user can perform	Actions to include in the policy
		Set retention policy	`DeleteRetentionPolicy` `PutRetentionPolicy`

Simple option for the trusted entity role	IAM	Create the `MediaLive AccessRole`	`CreateRole`
			`PutRolePolicy`
			`AttachRolePolicy`

		Choose the `MediaLive AccessRole`	`ListRole` `PassRole`

		Update the `MediaLive AccessRole`	`GetRolePolicy`
			`PutRolePolicy`
			`AttachRolePolicy`

Complex option for the trusted entity role	IAM	Enter a role for the trusted entity	`PassRole`

General activity that the user can perform	Corresponding service in IAM	Specific activities the user can perform	Actions to include in the policy

Deploy and work with AWS Elemental Link devices	MediaLive	Deploy, configure , and view an AWS Elemental Link device	`DescribeInputDevice`
			
			`DescribeInputDeviceThumbnail`
			
			`ListInputDevices`
			
			`RebootInputDevice`
			
			`StartInputDeviceMaintenanceWindow`
			
			`StartInputDevice`
			
			`StopInputDevice`
			
			`UpdateInputDevice`

General activity that the user can perform	Corresponding service in IAM	Specific activities the user can perform	Actions to include in the policy

Handle transfers of AWS Elemental Link devices	MediaLive	Handle transfers of AWS Elemental Link devices	`AcceptInputDeviceTransfer`
			`CancelInputDeviceTransfer`
			`ClaimDevice`
			`ListInputDeviceTransfers`
			`RejectInputDeviceTransfer`
			`TransferInputDevice`

Set up a AWS Elemental Link device as the source for a MediaConnect flow	MediaConnect	On the MediaLive console, view MediaConnect flows in the dropdown list. This dropdown list appears in the **Flow ARN** field in the **Attachments** tab on the **Device details** page.	`ListFlows`

General activity that the user can perform	Corresponding service in IAM	Specific activities the user can perform	Actions to include in the policy
	IAM	On the MediaLive console, view IAM roles in the dropdown list. This dropdown list appears in the **Role ARN** field in the **Attachments** tab on the **Device details** page.	`ListRoles`
	Secrets Manager	On the MediaLive console, view Secrets Manager secrets in the dropdown list. This dropdown list appears in the **Secret ARN** field in the **Attachments** tab on the **Device details** page.	`ListSecrets`
Set up email notification	Amazon SNS		All actions The managed policy `AmazonSNSFullAccess` provides these permissions

General activity that the user can perform	Corresponding service in IAM	Specific activities the user can perform	Actions to include in the policy

AWS Systems Manager	Systems Manager	Create a password parameter using the MediaLive console or the AWS Systems Manager console	`DeleteParameter`
			`DeletePar ameters`
			`DescribeP arameters`
			`GetParameter`
			`GetParame terHistory`
			`GetParameters`
			`GetParame tersByPath`
			`PutParameter`

	Systems Manager	Choose a password parameter from the dropdown list on the MediaLive console	`DescribeP arameters`

Requirements for AWS Elemental MediaLive features

You must give your users access to AWS Elemental MediaLive features. The permissions for MediaLive can be divided into three categories:

- Permissions to create
- Permissions to view
- Permissions to run

You might choose to give different access to different kinds of users. For example, you might decide that "basic operators" should not have create permissions.

In particular, you must decide whether to restrict the ability to work with reservations; you might decide to give this access only to administrators or advanced users. For more information about reservations, see the section called “Reservations”.

The following table shows the operations in IAM that relate to access for MediaLive.

Permissions	Service name in IAM	Actions

Create, modify, and delete channels, devices, inputs, and input security groups	MediaLive	`CreateChannel`
		`CreateInput`
		`CreateInputSecurityGroup`
		`DeleteChannel`
		`DeleteInput`
		`DeleteInputSecurityGroup`
		`UpdateChannel`
		`UpdateInput`
		`UpdateInputDevice`
		`UpdateInputSecurityGroup`

View channels, devices, inputs, and input security groups	MediaLive	`ListChannels`
		`ListInputDevices`
		`ListInputs`
		`ListInputSecurityGroups`

Permissions	Service name in IAM	Actions

		`DescribeChannel`
		
		`DescribeInput`
		
		`DescribeInputDevice`
		
		`DescribeInputDeviceThumbnail`
		
		`DescribeInputSecurityGroup`

View alerts for running channels Note that this action doesn't appear in the policy wizard on the IAM console. To include this action, create a policy, then edit the policy and type the line `"medialive:ListAlerts",` directly in the JSON. You can perform all these steps in the IAM console.	MediaLive	`ListAlerts`

Perform a batch operation on several channels or inputs or multiplexes or input security groups	MediaLive	`BatchDelete`
		`BatchStart`
		
		`BatchStop`

Permissions	Service name in IAM	Actions
Create or cancel an outgoing device transfer, or accept or reject an incoming device transfer, and view pending device transfers	MediaLive	`AcceptInputDeviceTransfer CancelInputDeviceTransfer ListInputDeviceTransfers RejectInputDeviceTransfer TransferInputDevice`
Work with schedules	MediaLive	`DescribeSchedule` `BatchUpdateSchedule`

Create or modify multiplexes	MediaLive	`CreateMultiplex`
		`DescribeMultiplex`
		`ListMultiplexes`
		`UpdateMultiplex`

	EC2	`DescribeAvailabilityZones` You need this operation to view the list of Availability Zones on the MediaLive console, so that you can choose two for the multiplex.

Delete multiplexes	MediaLive	`DeleteMultiplex`
		`DescribeMultiplex`
		`ListMultiplexes`

View multiplexes	MediaLive	`DescribeMultiplex` `ListMultiplexes`

Permissions	Service name in IAM	Actions
Change the class for a channel	MediaLive	`UpdateChannelClass`
Run channels	MediaLive	`StartChannel` `StopChannel`
Pause channels	MediaLive	Pause is part of the schedule feature, above.
Run multiplexes	MediaLive	`StartMultiplex` `StopMultiplex`

Attach tags to channels, inputs, and input security groups when creating those resources	MediaLive	`CreateTag`
		`DeleteTags`
		`ListTagsForResources`

Create, modify, delete, and view reservations and offerings	MediaLive	`DeleteReservation`
		`DescribeOffering`
		`DescribeReservation`
		`ListOfferings`
		`ListReservations`
		`PurchaseOffering`

Requirements for AWS CloudFormation

MediaLive includes a workflow wizard. Creation of a workflow always includes automatic creation of an AWS CloudFormation stack. Therefore, to use the workflow wizard, users need permissions in AWS CloudFormation.

Permissions	Service name in IAM	Actions

Work with the workflow wizard	AWS CloudFormation	`ListStacks`
		`DescribeStacks`
		`DescribeStackResources`
		`CreateStack`
		`DeleteStack`

Requirements for Amazon CloudFront

MediaLive includes a workflow wizard. One of the options in the wizard is to deliver output to AWS Elemental MediaPackage and from there to Amazon CloudFront. Therefore, for users to create a workflow with delivery to MediaPackage, users need permissions in CloudFront.

Permissions	Service name in IAM	Actions

Use the workflow wizard to create the CloudFront distribution that is associated with a MediaPackage channel, if your organization supports MediaPackage as an output destination.
Use the workflow wizard to delete a workflow that includes a CloudFront distribution.
	CloudFront	`ListDistributions`
		`DescribeDistribution`
		`CreateDistribution`
		`DeleteDistribution`

CloudFrontCreate and delete a CloudFront distribution, if your organization supports MediaPackage as an output destination.

Note how the required permissions here are very different from the permissions because the workflow wizard actually creates the distribution.

Requirements for AWS CloudTrail

MediaLive is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in MediaLive.

Users don't need special permissions for AWS CloudTrail.

Requirements for Amazon CloudWatch—monitoring channel health

The AWS Elemental MediaLive console includes a page (**Channel details**) that collects CloudWatch metrics information about the health of channels and displays it directly on the MediaLive console.

You must decide if you want to give some or all of your users permission to view metrics on the console.

For a user to view this information on the MediaLive console, that user must have view permissions for metrics operations in Amazon CloudWatch. When users have these permissions, they can also view the information through the CloudWatch console, AWS CLI, or REST API.

The following table shows the actions in IAM that relate to access for monitoring channel health.

Permissions	Service Name in IAM	Actions

View Metrics	CloudWatch	`ListMetrics`
		`GetMetricData`
		`GetMetricStatistics`

Requirements for CloudWatch and Amazon SNS—setting up email notification

MediaLive provides information about channels as they are running. It sends this information to Amazon CloudWatch as events. The details of these events can optionally be distributed to one or more users. Someone must set up this distribution. (For the setup procedure, see the section called “Monitor with CloudWatch events”.)

You must decide if you want to give some or all of your users these permissions. You might choose to allow each user to perform their own distribution setup. Or you might decide that an administrator must be responsible for performing the setup at startup for applicable users, and then again whenever a new user is added.

The following table shows the actions in IAM that relate to access for setting up email notification.

Permissions	Service Name in IAM	Actions
Write	CloudWatch Events	All actions
Write	SNS	All actions

Requirements for Amazon CloudWatch Logs—setting up channel logging

MediaLive produces channel logs that it sends to CloudWatch Logs, where users can view them. For more information about channel logs, see the section called “CloudWatch Logs”.

You must decide if you want to give some or all of your users permission to view the logs in CloudWatch Logs.

You must also decide if you want to give some or all of your users permission to set the retention policy for logs. If you decide not to give this access to any user, an administrator must be responsible for setting the policy.

Users don't need special permission to enable logging from within MediaLive.

The following table shows the actions in IAM that relate to access for setting up channel logs.

Permissions	Service name in IAM	Actions
View Logs	CloudWatch Logs	`FilterLogEvents` `GetLogEvents`
Set Retention Policy	CloudWatch Logs	`DeleteRetentionPolicy`

Permissions	Service name in IAM	Actions
		`PutRetentionPolicy`

Requirements for Amazon Elastic Compute Cloud—VPC inputs

Your deployment might include push inputs that connect to MediaLive from a VPC that you created with Amazon VPC.

When a user creates this type of input on the MediaLive console, they have the option to choose the subnet and security group from a dropdown list. For the dropdown list to be populated with the resources in Amazon VPC, the user must have the appropriate permissions. For more information about Amazon VPC inputs, see the section called “Creating an input”.

The following table shows the actions in IAM that relate to access for populating the dropdown.

Permissions	Service name in IAM	Actions

View the VPC subnets and VPC security groups on the MediaLive console	**EC2**	`DescribeSubnets`
		
		`DescribeSecurityGroups`

Requirements for Amazon Elastic Compute Cloud—delivery via VPC

Your deployment might include setting up some channels for delivery to output endpoints in Amazon Virtual Private Cloud (Amazon VPC).

When a user sets up for this feature on the MediaLive console, they have the option to choose subnets, security groups, and EIPs from a dropdown list. For the dropdown list to be populated with the resources in Amazon VPC, the user must have the appropriate permissions. For information about this feature, see the section called “VPC delivery”.

The following table shows the actions in IAM that relate to access for populating the dropdowns.

Permissions	Service name in IAM	Actions

View the VPC subnets and VPC security groups on the MediaLive console.	**EC2**	`DescribeSubnets`
		
		`DescribeSecurityGroups`

View the Elastic IP addresses on the console. The console finds the Elastic IP addresses that have been allocated for use in your AWS account.	**EC2**	`DescribeAddresses`

Requirements for IAM — trusted entity role

You must give your users access to a restricted set of IAM features, so that they can work with the trusted entity role. This role gives MediaLive the ability to perform operations on resources that belong to your organization's AWS account. MediaLive needs this ability when a channel is running.

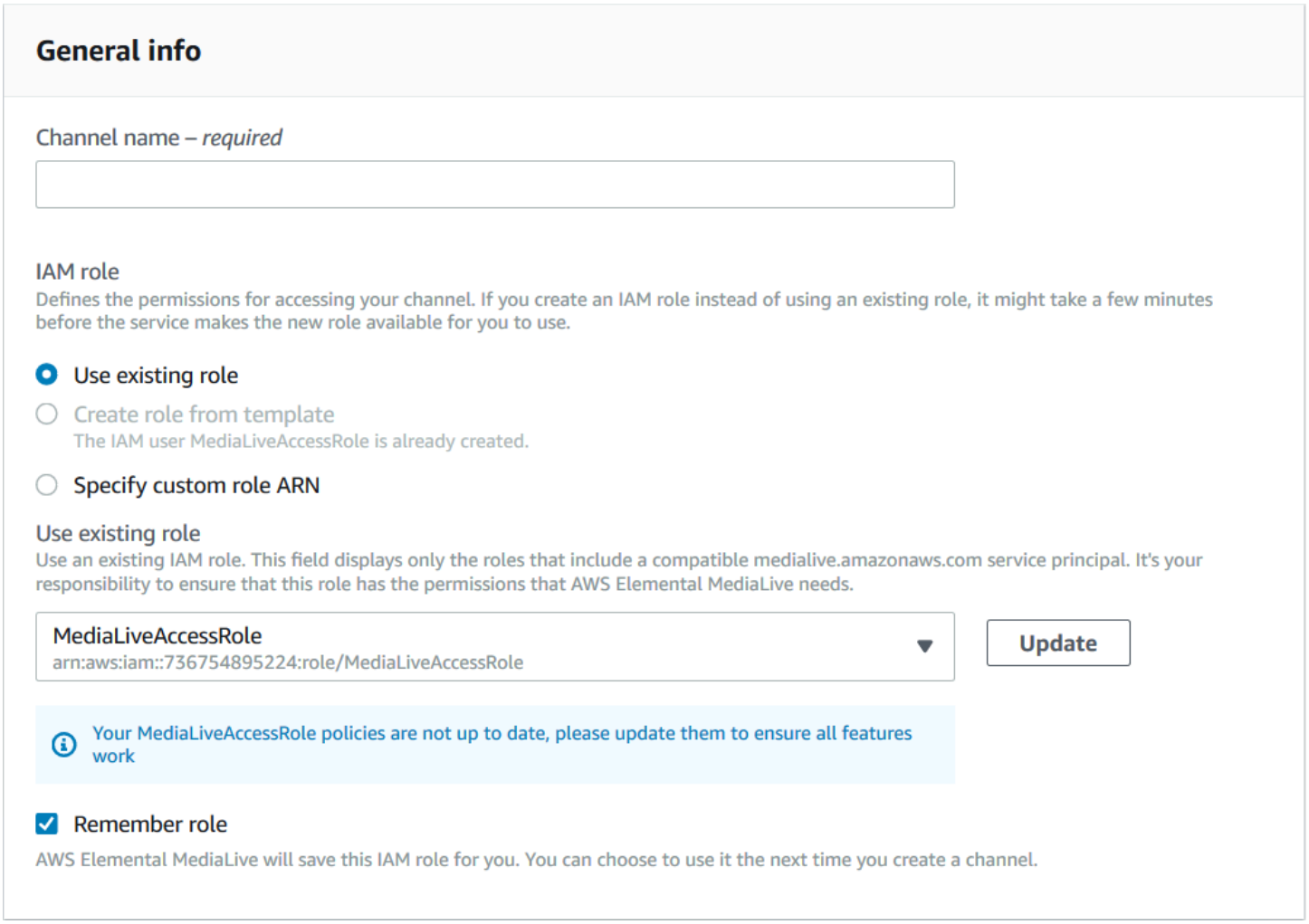
> **Important**
>
> Before you read this section, you should read the the section called “IAM permissions for trusted entity”, so that you understand how the trusted entity works, and so that you know whether your organization uses the simple option or the complex option.

User access for the simple option

Read this section if you decided that the simple option for the trusted entity is appropriate to your deployment. With this option, users work with trusted entity using only the MediaLiveAccessRole role.

Before you read this section, you should read the the section called “IAM permissions for trusted entity”, so that you understand how the trusted entity works, and so that you know whether your organization uses the simple option or the complex option.

For users to work in the **IAM Role** section on the **Channel and input details** pane, they must be able to select options on this pane. The following screenshot shows the **IAM Role** section as it appears when you start to create a channel.



You must give users the access described in the following table. All the actions are in the IAM service.

Field on the console	Description	Actions

Select Use existing role	Users must be able to select `MediaLiveAccessRole` from the selection field that accompanies the **Use existing role** field.	`ListRole`
		
		`PassRole`

Field on the console	Description	Actions

Select **Create role from template option**	Users must be able to select the **Create role from template** field.
	
	(The role needs to be created only once, by the first user to create a channel. But it is easiest to give these permissions to all users.)
		`CreateRole`
		
		`PutRolePolicy`
		
		`AttachRolePolicy`

Select **Specify custom role ARN**	Users don't need to be able to select this field. They will use `MediaLiveAccessRole` . They will never use a custom role.	None

Select **Update**	Users must be able to select this button so that MediaLive updates the `MediaLiveAccessRole` with new permissions. Permissions must sometimes be added to the role when a new feature is added to MediaLive.	`GetRolePolicy`
		
		`PutRolePolicy`
		
		`AttachRolePolicy`

User access for the complex option

Read this section if you decided that the complex option for the trusted entity is appropriate to your deployment. With this option, users work with trusted entity using the custom roles that you created.

Before you read this section, you should read the the section called “IAM permissions for trusted entity” and follow the procedure to create trusted entity roles for the complex option.

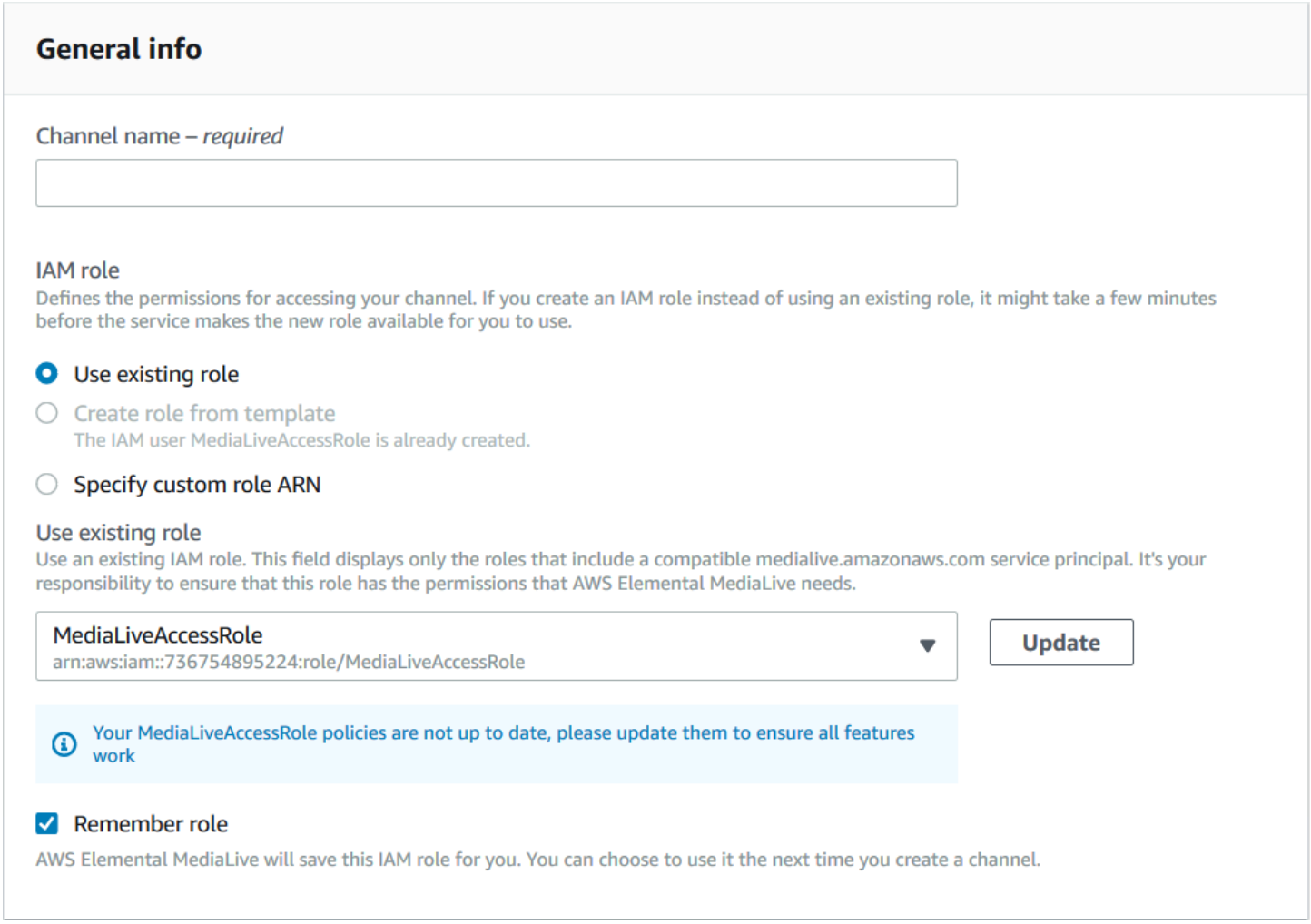
Topics

- Permissions that users need

- Information that users need

Permissions that users need

For users to work in the **IAM Role** section on the **Channel and input details** pane, they must be able to select options on this pane. The following screenshot shows the **IAM Role** section as it appears when you start to create a channel.



You must give users the access described in the following table. All the actions are in the IAM service.

Field on the console	Description	Actions
Select Use existing role	Users must *not* be able to view the list in the selection	None

Field on the console	Description	Actions
	field that accompanies the **Use existing role** field. That list shows all the roles that are created in the AWS account. Users must not be able to select from this list.	
Select **Create role from template option**	Users must *not* be able to select the **Create role from template** field. Users don't create roles. Only administrators create roles.	None
Select **Specify custom role ARN**	Users must be able to enter a role into the entry field that accompanies the **Specify custom role ARN** field. They must then be able to pass that role to MediaLive.	`iam:PassRole`
Select **Update**	Users do *not* need to be able to choose the **Update** button because this button only ever appears in implement ations that use `MediaLive AccessRole` . The complex option does not use this role; therefore, this button never appears.	None

Information that users need

When a user creates a channel, they will pass a role to MediaLive to set up MediaLive with the correct trusted policies. You created these policies when you set up the trusted entity. Specifically,

when you created the trusted entity role, you made a note of the ARNs of all the roles that you created.

You must give each user a list of the roles (identified by an ARN) that they must use with each workflow (channel) that they work with.

- Make sure that you give each user the correct roles for the workflows that they are responsible for. Each role gives MediaLive access the resources that apply for a specific workflow.
- Each user probably has a different list of roles.

When the user selects **Specify custom role ARN**, the user will consult their list to find the workflow the channel applies to and the role ARN that therefore applies.

Requirements for AWS Elemental Link

Your organization might deploy AWS Elemental Link hardware devices in one or both of these ways:

- As the video source for the input that you attach to an AWS Elemental MediaLive channel.
- As the video source for an AWS Elemental MediaConnect flow.

This section describes the permissions that you (an IAM administrator) must assign to users and other AWS identities so that they can configure an AWS Elemental Link device to work with a MediaLive input or an MediaConnect flow. For more information about these devices, see *Setup: AWS Elemental Link* .

Read this information as follows:

- Read this information if your organization has users who will both deploy devices and use those devices with MediaLive.
- Your organization might also have users who will only work with MediaLive to deploy devices and configure them for use as sources, and you might want to follow a *least permissions* rule for those users. If this is the case, see the section called “Setting up users with IAM permissions”.

You must assign permissions for actions in several services, as described in the following table.

Permissions	Service name in IAM	Actions

Deploy, configure, and view an AWS Elemental Link device	MediaLive	`DescribeInputDevice`
		`DescribeInputDeviceThumbnail`
		`ListInputDevices`
		`RebootInputDevice`
		`StartInputDeviceMaintenanceWindow`
		`StartInputDevice`
		`StopInputDevice`
		`UpdateInputDevice`

Handle transfers of AWS Elemental Link devices	MediaLive	`AcceptInputDeviceTransfer`
		`CancelInputDeviceTransfer`
		`ClaimDevice`
		`ListInputDeviceTransfers`
		`RejectInputDeviceTransfer`
		`TransferInputDevice`

On the MediaLive console, view MediaConnect flows in the dropdown list. This dropdown list appears in the **Flow ARN** field in the	MediaConnect	`ListFlows`

Permissions	Service name in IAM	Actions
Attachments tab on the **Device details** page.		
On the MediaLive console, view IAM roles in the dropdown list. This dropdown list appears in the **Role ARN** field in the **Attachments** tab on the **Device details** page.	IAM	`ListRoles`
On the MediaLive console, view Secrets Manager secrets in the dropdown list. This dropdown list appears in the **Secret ARN** field in the **Attachments** tab on the **Device details** page.	Secrets Manager	`ListSecrets`

Requirements for AWS Elemental MediaConnect

Your deployment might include using a flow from AWS Elemental MediaConnect as an input to AWS Elemental MediaLive.

Users need permissions to perform actions in MediaConnect when they use the MediaLive workflow wizard. Users don't need special permissions when they use the regular MediaLive console to specify a MediaConnect flow in an input or channel.

Permissions	Service name in IAM	Actions

Use the workflow wizard to create a MediaConnect flow, if your organization supports sources from MediaConnect.
 	
Use the workflow wizard to delete a workflow that
	MediaConnect	`List*`
		
		`Describe*`
		
		`Create*`
		
		`Delete*`

Permissions	Service name in IAM	Actions
includes a source from MediaConnect.		

Requirements for AWS Elemental MediaPackage

Your deployment might send outputs to AWS Elemental MediaPackage, either by creating an HLS output group or by creating a MediaPackage output group. (Note that both MediaLive and MediaPackage have "channels"; however, they are different objects.)

The user needs permissions to perform actions in MediaPackage when they use the MediaLive console and when they use the MediaLive workflow wizard.

Permissions	Service name in IAM	Actions
On the MediaLive console, view the MediaPackage channels in the dropdown list on the MediaLive channel.	MediaPackage	`Describe*`

Use the workflow wizard to create a MediaPackage channel, if your organization supports MediaPackage as an output destination.
 	
Use the workflow wizard to delete a workflow that includes a MediaPackage output.
	MediaPackage	`List*`
		
		`Describe*`
		
		`Create*`
		
		`Delete*`

Requirements for AWS Elemental MediaStore

Your deployment might include using files in an AWS Elemental MediaStore container. For example, your deployment might use files in the following ways:

- As the source for an HLS input
- As the destination for an HLS output group

The user needs permissions to perform actions in MediaStore when they use the MediaLive workflow wizard. The user doesn't need special permissions when they use the regular MediaLive console to specify a MediaStore container in a channel.

Permissions	Service name in IAM	Actions

Use the workflow wizard to create a MediaStore container , if your organization supports MediaStore as an output destination.
 	
Use the workflow wizard, to delete a workflow that includes a MediaStore output.
	MediaStore	`List*`
		
		`Describe*`
		
		`Create*`
		
		`Delete*`

Requirements for AWS Resource Groups—tagging

When users create channels, inputs, or input security groups, they can optionally attach tags to the resource during creation. Typically, your organization has a policy to tag or to omit tags. There are two services that control permissions for tagging, for two different scenarios:

- The ability to tag during channel creation is controlled by actions within AWS Elemental MediaLive. See the section called “MediaLive”.
- The ability to modify tags in existing resources is controlled by actions within Resource Group Tagging. See Working with Tag Editor in Getting Started with the AWS Management Console.

Requirements for Amazon S3

Your deployment might include using files in an Amazon S3 bucket. For example, your deployment might use files in the following ways:

- As the source for an HLS input

- As the destination for an Archive output group
- As the destination for an HLS output group

Users don't need special permissions to specify an Amazon S3 bucket in a field on the MediaLive console.

Requirements for AWS Systems Manager—creating password parameters in parameter store

The AWS Elemental MediaLive console includes a feature that lets a user create a password parameter in the AWS Systems Manager Parameter Store. This feature is part of the **Create Channel** page. This feature does not exist in the AWS CLI or REST API.

You must decide if you want to give some or all of your users permission to use this feature. (If you don't give this access to any users, then an administrator must be responsible for creating parameters.)

About the feature for creating password parameters

The AWS Systems Manager Parameter Store is used extensively in AWS Elemental MediaLive. It is likely that you will use this store. The store holds passwords that MediaLive needs to retrieve and store files externally.

Here are some of the MediaLive functions that use this store to hold passwords:

- An input of type RTMP Pull or type HLS Pull, if the connection is secure.
- Fields in the channel that hold the URL to an external file, if the connection is secure. An example of this type of field is **Avail blanking image**.
- The destination in an HLS output group or a Microsoft Smooth output group, if the connection is secure.

In all these cases, MediaLive needs the user name and the password. The password is always stored in a parameter. Therefore, the console includes a **Username** field and a **Password parameter** field. For an example of the relevant fields, open the MediaLive console, choose **Create channel**, **General settings**, **Avail blanking**, **Avail blanking image**, and then choose **Credentials**.

How password parameters work

The password parameter feature ensures that when the user is creating a channel, AWS Elemental MediaLive does not store passwords in plaintext. It works as follows:

- First, a user or administrator creates a password parameter in AWS Systems Manager Parameter Store. The parameter is a name-value pair where the name is something like **`corporateStorageImagesPassword`** and the value is the actual password.
- Second, when a user is creating a channel or input in MediaLive and needs to enter a password, the user specifies the password parameter name instead of the password. That name is stored in MediaLive. The actual password is never stored in MediaLive.
- Finally, when the channel is running and MediaLive needs the password (to either read or write to the external location), it sends the password parameter name to Parameter Store and gets back the actual password in response.

Create feature that is built into AWS Elemental MediaLive

When a password field appears on the console, AWS Elemental MediaLive includes a feature that lets the user do one of the following:

- Enter the name of an existing password parameter.
- Create a password parameter by entering the name-value pair (a parameter name and an actual password).

Required permissions

Users must enter the name of a password parameter or select a name from the dropdown list. Some users might need permission to create a password parameter within AWS Elemental MediaLive.

Permission to enter a name

No special permission is required to enter the name of an existing password parameter on the AWS Elemental MediaLive console.

Permission to select a name

For the user to select a name from the dropdown list, the user must have permission for `GetParameters` in AWS Systems Manager.

Permission to create

For any user to create a password parameter on the AWS Elemental MediaLive console, that user must have permission to specific operations in AWS Systems Manager Parameter Store. (With this permission, the user can also create these password parameters ahead of time on the AWS Systems Manager console. The user can choose the option that they prefer.)

You can give access to some or all users to create these password parameters. Typically, you give this access only to users who are trusted with sensitive passwords; these might be users whom you have identified as advanced users:

- If you give access only to advanced users, those users must be responsible for creating parameters at startup for the applicable assets and whenever a new asset is required by MediaLive. The users can perform the setup on the MediaLive console or on the AWS Systems Manager console.
- If you don't give this access to any users, an administrator must be responsible for creating parameters at startup for the applicable assets and whenever a new asset is required by MediaLive. An administrator might prefer to perform this setup on the AWS Systems Manager console.

Permission to modify and delete

If you want users to be able to modify and delete password parameters (as well as create them), give access to modify and delete operations. The users will be able to modify and delete from the AWS Systems Manager Parameter Store. (There is no feature on the AWS Elemental MediaLive console for modifying and deleting.)

You might choose to give this access to the users who have create permissions. Or you might choose to give this access only to administrators.

The following table shows the actions in IAM that relate to access for the Parameter Store.

Permissions	Service name in IAM	Actions
Select	Systems Manager	`GetParameters`
Create	Systems Manager	`PutParameter`
Modify and Delete	Systems Manager	`DeleteParameter`

Permissions	Service name in IAM	Actions
		DeleteParameters DescribeParameters GetParameter GetParameterHistory GetParameters GetParametersByPath

IAM permissions for MediaLive as a trusted entity

AWS Elemental MediaLive must be set up so that when a channel is running, MediaLive itself has access to perform operations on resources that belong to your organization's AWS account. In other words, MediaLive must be set up as a *trusted entity* on your organization's account.

This section describes how to perform three setups related to the trusted entity:

- Determine the option to follow to implement the trusted entity.
- Set up MediaLive as a trusted entity, and set up the users so that they can work with the trusted entity when they work in MediaLive.

About the trusted entity role

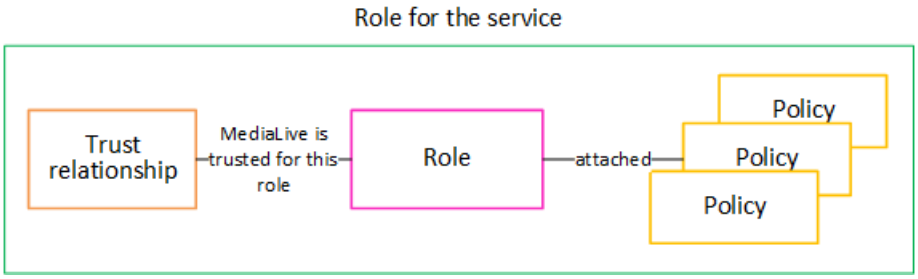
AWS Elemental MediaLive must be set up so that when a channel is running, MediaLive itself has access to perform operations on resources that belong to your organization's AWS account. For example, your deployment might use Amazon S3 as a source for files, such as blackout images, that MediaLive requires during processing. For MediaLive to obtain these files, it must have read access to some or all buckets in Amazon S3.

To perform the required operations on those resources, MediaLive must be set up as a *trusted entity* on your account.

MediaLive is set up as a trusted entity as follows: A role (that belongs to your AWS account) identifies MediaLive as a trusted entity. The role is attached to one or more policies. Each policy

contains statements about allowed operations and resources. The chain between the trusted entity, role, and policies makes this statement:

"MediaLive is allowed to assume this role in order to perform the operations on the resources that are specified in the policies."



After this role is created, the MediaLive user attaches the role to a specified channel, when they create or edit the channel. This attachment makes this statement:

"For this channel, MediaLive is allowed to assume this role in order to perform the operations on the resources specified in the policies."

The attachment is at the channel level, which gives the flexibility of creating different roles for different channels. Each role gives MediaLive access to different operations and, especially, different resources.

Step 1: Choose the option for implementing the trusted entity

There are two options for setting up the trusted entity role in AWS Elemental MediaLive: a simple option and a complex option.

Your organization must decide which option to use. This decision must be made by a person in your organization who understands your organization's requirements for access to resources. This person must understand whether there is a requirement that AWS Elemental MediaLive channels should be restricted in their access to resources in other AWS services. For example, this person should determine whether channels should be restricted in their access to buckets in Amazon S3 so that a specified channel can access some buckets and not others.

Topics

Simple option

The simple option typically applies when both these situations apply:

- Users in your organization are using AWS Elemental MediaLive to encode the organization's own assets (not assets belonging to customers).
- Your organization doesn't have rigorous rules about accessing assets. For example, you don't have video assets that can be handled only by specific users or departments.

With the simple option, there is only one role: `MediaLiveAccessRole`. All channels use this role and all users in your organization can attach that role to the channels that they work with.

The `MediaLiveAccessRole` role grants broad access to operations and complete access to all resources. It allows either read-only access or read/write access to all the services that MediaLive must access when a channel is running. And most significantly, it allows full access to all the resources associated with those services.

If the simple option is suitable to your deployment, follow the steps in the section called “Step 2: Set up with simple option”.

Complex option

The complex option applies when the `MediaLiveAccessRole` role is too broad for your use, given that it allows broad access to operations and complete access to all resources.

For example, you might have the following requirements:

- A requirement that a given channel should be allowed to access only specific resources, and another channel should be allowed to access only specific, different resources. In a situation like this, you need to create several access roles. Each role narrows down permissions to a different set of resources.
- A requirement that each user should be allowed to display only specific roles on the console, to prevent a user from viewing a role they should not know about or to prevent a user from selecting the wrong role. For example, you might want to set up so that only user A can work with workflow X, and you might further require that only user A knows about workflow X.

If the complex option is applicable to your deployment, follow the steps in the section called “Step 2: Set up with complex option”.

Step 2: Create the trust entity – simple option

Read this section if you decided that you should use the simple option for setting up the trusted entity.

With this option, one of the MediaLive users creates the trusted entity using a wizard in the MediaLive console. The wizard provides a mechanism to create the trusted entity. It also includes a mechanism to update the trusted entity with new permissions. Permissions must sometimes be added to the trusted entity when a new feature is added to MediaLive.

Tasks for you to perform

Therefore, with this option, there is only one task for you to perform. You must set up all MediaLive users with permissions so any of these users can use the wizard to perform two types of activities:

- A user can create and update the trusted entity.
- A user can use the wizard to attach a specific trusted entity to a channel.

For detailed information about this setup, see the section called "User access for the simple option".

Step 2: Create the trusted entity — complex option

Read this section if you decided that you should use the complex option for setting up the trusted entity.

You must identify the services that MediaLive will interact with in your deployment. Then within each service, you must identify the operations and resources that MediaLive needs access to. You must then create policies to describe the required access, create a role for MediaLive, and create a trust relationship (a trusted entity) that attaches the policies to the role. For a diagram of how the policies, role, and trusted entity work together, see the section called "About the trusted entity role".

After you have set up the trusted entity, the users of MediaLive will select the trusted entity to attach to the channel that they are creating.

Tasks for you to perform

Therefore, with this option, you must perform these tasks:

- Create policies and roles, and use those policies and roles to set up MediaLive as a trusted entity.

- Set up all MediaLive users with permissions that let them attach a specific trust policy to a channel, when they create or edit the channel.

These steps are described in steps A to E.

Topics

Step A: Determine the access requirements

You must identify the services that MediaLive will interact with in your deployment. Then within each service, you must identify the operations and resources that MediaLive needs access to. Finally, you must design the IAM policies that handle these requirements.

This requirements analysis must be performed by a person in your organization who understands your organization's requirements for access to resources. This person must understand whether there is a requirement that MediaLive channels should be restricted in their access to resources in other AWS services. For example, this person should determine whether channels should be restricted in their access to buckets in Amazon S3 so that a specified channel can access some buckets and not others.

To determine the access requirements for MediaLive

1. See the table in the section called “Reference: summary of trusted entity access” for information about the services that MediaLive typically needs access to. Determine which of those services your deployment uses and which operations it needs.
2. Within a service, determine the number of policies that you need to create. Do you need several different combinations of objects and operations for different workflows, and do you need to keep those combinations separate from each for security reasons?

   Specifically, determine whether you need access to different resources for different workflows, and whether it's important to restrict access to specific resources. For example, in AWS Systems Manager Parameter Store you might have passwords that belong to different workflows, and you might want to allow only specific users to access the passwords for any given workflow.

If different workflows have different requirements for objects, operations, and resources, then for that service you need separate policies for each workflow.

3. Design each policy: identify the allowed (or not allowed) objects, operations, and the allowed (or not allowed) resources in the policy.
4. Determine if any of the policies that you have identified are covered by a managed policy.
5. For each workflow, identify the policies that you need for all the services that the workflow uses. When you create the policy, you will be able to include several services in the policy. You don't need to create a policy for each separate service.
6. Identify the number of roles that you need. You need one role for each unique combination of policies.
7. Assign names to all the policies and roles that you have identified. Take care not to include sensitive identifying information (such as a customer account name) in these names.

Step B: Create policies

After you have identified the policies that you need, create them on the IAM console. Follow this procedure for each policy. Remember that this policy is attached to the MediaLive trusted entity that the user chooses. You should only include actions that you want MediaLive to be able to perform at runtime when working on behalf of that user.

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane on the left, choose **Policies**. Then choose **Create policy**. The **Create policy** wizard appears. This wizard walks you through the steps, including these key steps:
   - Select a service.
   - Select actions for that service.

     Typically (and by default), you specify the actions that you want to allow.

     But you can also choose the **Switch to deny permissions** button to deny the chosen actions instead. We recommend as a security best practice that you deny permissions only if you want to override a permission separately allowed by another statement or policy. We recommend that you limit the number of deny permissions to a minimum because they can increase the difficulty of troubleshooting permissions.

- Specify resources for each action (if supported for the action). For example, if you choose the MediaLive `DescribeChannel` ARN you can specify the ARNs of specific channels.
- Specify conditions (optional). For example:
  - You can specify that a user is allowed to perform an actions only when that user's request happens within a certain time range.
  - You can specify that the user must use a multi-factor authentication (MFA) device to authenticate.
  - You can specify that the request must originate from a range of IP addresses.

  For lists of all of the context keys that you can use in a policy condition, see Actions, resources, and condition keys for AWS services in the *Service Authorization Reference*.

3. Choose **Create policy**.

Step C: Create roles

Any person who is an administrator can perform the procedure to create a role and attach policies to the role.

In the section called "A: Determine requirements", someone in your organization identified the roles that you need to create. Create those roles now using IAM.

In this step, you create a role that consists of a trust policy ("let MediaLive call the `AssumeRole` action") and one or more policies (the policies that you just created). In this way, MediaLive has permission to assume the role. When it assumes the role, it acquires the permissions specified in the policies

1. On the IAM console, in the navigation pane on the left, choose **Roles**, then **Create Role**. The **Create role** wizard appears. This wizard walks you through the steps of setting up a trusted entity, and adding permissions (by adding a policy).
2. On the **Select trusted entity** page, choose the **Custom trust policy** card. The **Custom trust policy** section appears, with a sample policy.
3. Erase the sample, copy the following text, and paste the text in the **Custom trust policy** section. The **Custom trust policy** section now looks like this:

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
 {
            "Effect": "Allow",
            "Principal": {
                "Service": "medialive.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

4. Choose **Next**.
5. On the **Add Permissions** page, find the policy or policies that you created (for example, `medialiveForLinkFlowAccess`), and select the checkbox for each. Then choose **Next**.
6. On the review page, enter a name for the role. We recommend that you don't use the name `MediaLiveAccessRole` because it is reserved for the simple option. Instead, use a name that includes `medialive` and describes this role's purpose.
7. Choose **Create role**.
8. On the **Summary** page for the role, make a note of the value in **Role ARN**. It looks like this:

   `arn:aws:iam::111122223333:role/medialiveWorkflow15`

   In the example, `111122223333` is your AWS account number.
9. Make a list of the ARNs for all the roles you create. Include the following information in each item:
   - The role ARN
   - A description of the workflow that the ARN applies to.
   - The users who can work with this workflow and therefore need the ability to attach this trust policy to the channels that they create and edit.

   You will need this list when you set up trusted entity access for users.

Step D: Set up users

After you have set up all the trust policies that your organization requires, you must set up each MediaLive user (or group of users) with permissions. These permissions let each user attach specific trust policies to a channel. For more information, see the section called “User access for the complex option”.

Reference: summary of requirements for the MediaLive trusted entity

The following table shows all the types of permissions that the MediaLive trusted entity might need. Refer to this table when you determine the access requirements for the MediaLive trusted entity.

Each row in the column describes a task or set of related tasks that the MediaLive trusted entity might need to perform for a user. The third column describes the type of access that the trusted entity requires to perform that task. The last column lists the IAM actions or policy that control that access.

Service	Tasks	Type of access required	Suggested actions or policy
AWS Elemental MediaLive	Working with MediaLive features.	MediaLive doesn't need access to itself. Only the users need access.	
AWS CloudTrail	Capturing MediaLive activity.	MediaLive doesn't need IAM access for this task.	
CloudWatch	Displaying CloudWatch metrics information on the console, to monitor channel health.	MediaLive doesn't need IAM access for this task. Only the users need access.	
CloudWatch Events and Amazon SNS	Setting up email notification so that users can be notified about MediaLive alerts that are sent to CloudWatch Events.	MediaLive doesn't need access for this task. Only the users need access.	
CloudWatch Logs	Sending channel log information to	When the channel is running.	`CreateLogGroup` `CreateLogStream`

Service	Tasks	Type of access required	Suggested actions or policy

	CloudWatch Logs when a channel is running.	MediaLive must be able to send log messages to CloudWatch Logs .	`PutLogEvents`
			
			`PutMetricFilter`
			
			`PutRetentionPolicy`
			
			`DescribeLogStreams`
			
			`DescribeLogGroups`
			
			And these resources:
			
			`arn:aws:logs: *`
			
			`arn:aws:log-group: *`

Amazon EC2	Creating a CDI VPC, an RTP VPC input, or an RTMP VPC push input.	When the user is creating a VPC input.
		
		MediaLive must have write access to Amazon EC2 in order to create network interfaces for the input.
			`CreateNetworkInterface`
			
			`CreateNetwork Interface Permission`
			
			`DescribeNetworkInterfaces`
			
			`DescribeSecurityGroups`
			
			`DescribeSubnets`

Service	Tasks	Type of access required	Suggested actions or policy

	Deleting a CDI VPC, an RTP VPC input, or an RTMP VPC push input.	When the user deletes a VPC input.
		
		MediaLive must have write access to Amazon Elastic Compute Cloud in order to delete the network interfaces for the input.
			`DeleteNetworkInterface`
			
			`DeleteNetworkInterfacePermission`
			
			`DescribeNetworkInterfaces`
			
			`DescribeSubnets`

	Setting up a channel for delivery of output via your VPC	Create and delete elastic network interfaces on your VPC.
		
		MediaLive creates these network interfaces in the subnet for the channel pipeline endpoints.
			`CreateNetworkInterface`
			
			`CreateNetworkInterfacePermission`
			
			`DeleteNetworkInterface`
			
			`DescribeSubnets`
			
			`DescribeSecurityGroups`
			
			`DescribeAddresses`

Service	Tasks	Type of access required	Suggested actions or policy

		Associate Elastic IP addresses with the elastic network interfaces that MediaLive creates. Associating Elastic IP addresses is optional.
		
		There is no need to give access to `DisassociateAddress` . When MediaLive deletes any unnecessary network interfaces, the Elastic IP address will be automatically disassociated from the network interface .
			`AssociateAddress`
			
			`DescribeAddresses`

Service	Tasks	Type of access required	Suggested actions or policy

AWS Elemental MediaConnect	Creating a MediaConnect input.	When the user creates a MediaConnect input.
		
		MediaLive must have read/write access to the MediaConnect flow, in order to add an output to that flow.
			`ManagedDescribeFlow`
			
			`ManagedAddOutput`
			
			To include these actions that start with "Managed" in a policy, you must view the policy in the **JSON** tab and enter the names of the actions. You can't use the **visual editor** to choose these actions.

	Deleting a MediaConnect input.	When the user deletes a MediaConnect input.
		
		MediaLive should have read/write access to the MediaConnect flow, in order to delete the outputs on the flow, because the outputs are no longer needed.
			`ManagedDescribeFlow`
			
			`ManagedRemoveOutput`
			
			To include these actions that start with "Managed" in a policy, you must view the policy in the **JSON** tab and enter the names of the actions. You can't use the **visual editor** to choose these actions.

Service	Tasks	Type of access required	Suggested actions or policy
	Creating a MediaConnect entitlement. When the user creates a multiplex, MediaLive automatically creates an entitlement as the destination for the MPTS.	MediaLive doesn't need access for this task.	
AWS Elemental MediaPackage	Sending channel output to MediaPackage when a channel is running, if your deployment uses this service.	When the user creates a MediaPackage output group. MediaLive must have read access to the AWS Elemental MediaPackage channel, in order to obtain the credentials required to send to that channel.	`DescribeChannel`

Service	Tasks	Type of access required	Suggested actions or policy
	Sending channel output to MediaPackage v2 when a channel is running, if your deployment uses version 2 of that service. To deliver in this way, you create an HLS output group, not a MediaPackage output group.	When the channel is running. When the channel includes an HLS output that is delivering to a MediaPackage channel that uses MediaPackage v2. MediaLive must have write access to the AWS Elemental MediaPackage channel.	`mediapackagev2:PutObject`

AWS Elemental MediaStore	Sending and retrieving assets from a MediaStore container when a channel is running, if your deployment uses this service.	When the channel is running.
		
		MediaLive must have read access (for a source) or read/write access (for a destination).
			`ListContainers`
			
			`DescribeObject`
			
			`PutObject`
			
			`GetObject`
			
			`DeleteObject`

Resource Group Tagging	Attaching tags when creating resources—channels, inputs, and input security groups—and revising tags on existing resources.	MediaLive doesn't need IAM access for this task. Only the users need access.	

Service	Tasks	Type of access required	Suggested actions or policy

Amazon S3	Sending and retrieving assets from an Amazon S3 bucket when a channel is running, if your deployment uses this service.	When the channel is running.
		
		MediaLive must have read access (for a source) or read/write access (for a destination) to the buckets.
			`ListBucket`
			
			`PutObject`
			
			`GetObject`
			
			`DeleteObject`

	Sending thumbnails to an Amazon S3 bucket when a channel is running, if a channel has input thumbnails enabled	When the channel is running. MediaLive must have read/write access.	`PutObject`
AWS Systems Manager	Creating a password parameter on the MediaLive console.	MediaLive doesn't need IAM access for this task. Only the users need access.	
	Using a password parameter in the channel configuration. See the section called “AWS Systems Manager parameter store”.	When the channel is running. MediaLive must have read access to the AWS Systems Manager Parameter Store.	The managed policy `AmazonSSMReadOnlyAccess`

Ways to work with AWS Elemental MediaLive

Here are three suggestions about how to get started with AWS Elemental MediaLive.

- Use the workflow wizard to quickly create a functioning channel. See the section called "Workflow wizard".

  The wizard provides a compact user experience and supports some easy and popular input (source) types and output types.

  If you are new to video encoding, the wizard can get you started, and might be all that you need.

  If you are already experienced with video encoding, the wizard lets you quickly create a basic workflow that you can then enhance, using the regular MediaLive console.
- Follow the tutorial.

  You might follow the wizard to create a workflow, then decide that you need to add more features to the workflow. The tutorial provides an introduction to the basic workings of the MediaLive console.
- Start designing a workflow, either from scratch, or to enhance the workflow wizard. In this case, you should read the guide, starting with the sections that are prefixed with *Setup*. These sections provide detailed instructions on designing your workflow, and designing the channel inside that workflow. These sections contain cross references to the remaining sections in the guide.

Topics

Using the MediaLive workflow wizard

The AWS Elemental MediaLive workflow wizard lets you quickly get an MediaLive channel up and running. The wizard creates a channel and inputs (if they don't already exist). But it also creates related resources in other services (such as a channel in AWS Elemental MediaPackage). So it helps with the entire workflow, not just with the MediaLive portion.

Topics

About the workflow wizard

Supported inputs

To use the workflow wizard, you must ingest a single source from one of the following:

- A flow in AWS Elemental MediaConnect.
- Content from an AWS Elemental Link hardware device.
- Content from a mobile phone or webcam. The source is delivered using the RTMP protocol.
- An MP4 file stored on Amazon S3 or an HTTP server.

Supported outputs

With the workflow wizard, you can extract one video asset and one audio asset from the source and convert it to one or more of the following types of output:

- Output to send to an AWS Elemental MediaPackage channel, for delivery to Amazon CloudFront. CloudFront distributes the content to your end users.
- Output to send to an AWS Elemental MediaStore container, for delivery to CloudFront. CloudFront distributes the content to your end users.
- Output to Facebook, Twitch, or YouTube.

Low-touch setup

The workflow wizard automatically performs as much setup as possible in the upstream system. For example, your source might be in MediaConnect but you might not have created the flow. In this case, the workflow wizard automatically creates the flow for you.

The workflow wizard also automatically performs as much setup as possible in the downstream system or systems. For example, you might be sending to MediaPackage, but you might not

have created the MediaPackage channel and might not have set up CloudFront. In this case, the workflow wizard performs the setup in those services.

The workflow wizard works best when you use it to create these resources, rather than using existing resources.

Using the workflow wizard

Creating a workflow

1. Determine which type of output or outputs you need.

   If you're not sending to Facebook, Twitch, or YouTube, your major decision is whether to use MediaPackage or MediaStore. If you plan to repackage the output, choose MediaPackage. If you don't know about repackaging and suspect that you don't need it, you could choose MediaStore. You can always modify the workflow later, if you find you made the wrong decision.
2. Determine which type of source you have. If necessary, speak to the person who is providing the source.
3. If the source is using the RTMP protocol, you must set up an input security group using the regular MediaLive console. See the section called “Input security groups”.
4. Make sure that you have set up the IAM permissions that your users must have so that they can create and run the workflow. See the section called “IAM permissions for users”, and specifically the section called “Reference: summary of user access”
5. Sign in to the AWS Management Console and open the MediaLive console at https://console.aws.amazon.com/medialive/.
6. Choose **Workflow wizard** from the navigation panel. Follow the steps in the workflow wizard.
7. After you choose **Create** on the page, details about the workflow appear. A card appears for each resource that the workflow wizard involves.

   The workflow wizard creates a AWS CloudFormation stack. AWS CloudFormation runs that stack to create all the other resources:

   - One MediaLive input.
   - One MediaLive channel.
   - All the resources in all AWS services that are involved in the workflow you have created. You workflow might involve MediaPackage, MediaStore, and CloudFront.

8. When the resources have all been created, you can choose **Start workflow** on the details page for the workflow. The wizard starts the channel. The wizard also starts the MediaConnect flow, if you have one.

Modifying a workflow

You can't use the workflow wizard to modify an existing workflow. For suggestions about making changes, see the section called "Next steps—novice users" and the section called "Next steps—experienced video users".

Deleting a workflow

You can delete the workflow. MediaLive handles the resources that belong to the workflow as follows:

- It always deletes the channel.
- It always deletes the AWS CloudFormation stack.
- It deletes the input, if the workflow wizard created it. It doesn't delete the input if the input already existed.

- It deletes the flow associated with a MediaConnect input (if there is one), if the workflow wizard created the flow.
- It deletes the MediaPackage channel (if there is one) and its endpoints.
- It attempts to delete the MediaStore container, if there is one and if the workflow wizard created it. Deletion will fail if the container has any objects in it, including objects that aren't associated with this workflow.
- It deletes the CloudFront distribution, if the workflow wizard created it.

Next steps—novice users

If you are new to the world of video streaming and have fairly modest requirements, you might find that the workflow wizard implements all the features you need, and that the **Workflow Details** page gives you the monitoring details and runtime controls you require.

But if you want, you can use the regular MediaLive console to add more features to the channel. For example, you could add captions to the output (assuming that the source includes captions).

See the section called "Editing and deleting a channel" for information about modifying the channel.

If you created a workflow that involves MediaStore, MediaPackage, and CloudFront, you should read the user guides for those services, to better understand their roles, and for information on the features of those services that you could add.

You should also read the information on pricing for MediaLive, and for other AWS services, so that you understand the AWS charges that your workflow incurs. For information about MediaLive charges, see the section called "Pricing".

Next steps—experienced video users

If you have experience with video streaming and with other AWS services, you might want to add more MediaLive resources and more resources from other AWS services to the workflow. Following are some of the ways that you can revise the workflow.

- You can work with each service, using the AWS console or an AWS SDK. For example, you can use the MediaLive console to add more MediaLive inputs to the channel. Or you can use the AWS CLI to create a MediaStore container and then create a new MediaLive output in your channel that uses that container as a destination.
- You can use AWS CloudFormation to revise the AWS CloudFormation stack, to include more resources for AWS CloudFormation to create. For example, you could create and attach more MediaLive inputs. Or you could add an AWS Lambda function to the workflow. For more information, display the details page for the workflow in the MediaLive console, then choose the appropriate link.
- You can use the Media Services Application Mapper (MSAM) to monitor your resources. For more information, display the details page for the workflow in the MediaLive console, then choose the appropriate link.

If you created a workflow that involves MediaStore, MediaPackage, and CloudFront, you should read the user guides for those services, to better understand their roles, and for information on feature of those services that you could add.

You should also read the information on pricing for MediaLive, and for other AWS services, so that you understand the AWS charges that your workflow incurs. For information about MediaLive charges, see the section called "Pricing".

AWS Elemental MediaLive tutorial

This tutorial describes how to ingest a video source from an RTP source and generate one HLS output that contains one H.264 video encode and one audio encode. MediaLive will send the output to AWS Elemental MediaPackage. The output will consist of the following:

- One parent manifest: channel.m3u8
- One rendition manifest: channel-1.m3u8
- TS files for each output: channel-1.00001.ts, channel-1.00002.ts, channel-1.00003.ts, and so on

This tutorial uses the default values for most configuration fields in the channel.

> **Note**
>
> All the text marked as an example in this tutorial is just that—a sample that shows what a piece of information typically looks like. You must replace each example with the information that is valid for your situation.

Topics

Prerequisites

Before you can use MediaLive, you need an AWS account and the appropriate permissions to access, create, and view MediaLive components. Complete the steps in the section called "Preliminary setup steps", and then return to this tutorial. You can't use MediaLive, even as an administrator with full access, until you perform those steps.

Step 1: Set up the upstream system

The upstream system is the system that streams the video to MediaLive. The upstream system can be anything from an on-premises appliance that is serving as a "contribution encoder" to an application running on a smart phone. You must perform some setup of your upstream system before you start working with MediaLive.

For the purposes of this tutorial, the upstream system must be capable of sending a video stream via RTP push.

In a "push" delivery, the upstream system is pushing the stream *from* two IP addresses on the upstream system (for example, from **`203.0.113.111`** and from **`203.0.113.112`**). The upstream system will push *to* two IP addresses on MediaLive (for example, **`rtp://198.51.100.10:5000`** and **`rtp://192.0.2.131:5000`**). In the following steps, you will set up MediaLive so that the two *from* IP addresses are white listed. Furthermore, MediaLive will generate the two *to* IP addresses. You will set up the upstream system to push to those addresses.

To set up the upstream system

1. Set up your upstream system to perform an RTP push from two different IP addresses. You must push from two addresses because MediaLive always expects redundant inputs.
2. Make a note of the IP addresses. For example, **`203.0.113.111`** and from **`203.0.113.112`**. You will need these addresses when you set up the input security group in a later step.

Step 2: Set up the downstream system

In this tutorial, the downstream system (the destination for the output from MediaLive) is AWS Elemental MediaPackage.

You must set up a channel in AWS Elemental MediaPackage, and you must set it up now because you need the two input URLs that AWS Elemental MediaPackage generates. You enter these input URLs into MediaLive.

To set up the downstream system

1. Sign in to the AWS Management Console and open the MediaPackage console at https://console.aws.amazon.com/mediapackage/.
2. In a new web browser tab or window, display the Getting Started for AWS Elemental MediaPackage and follow steps 1 to 3 to create one channel and its endpoint.
3. Make a note of the data that AWS Elemental MediaPackage has generated: two input URLs and their associated names and passwords. For example, the data for one input URL might be:

   - `https://39fuo4.mediapackage.us-east-1.amazonaws.com/in/v1/88dpie/channel`
   - `ue739wuty`
   - `due484u`

   Your channel might be in a different Region from the example.
4. Keep the web browser open; don't close it yet.

Step 3: Create an input

You must create an input. the input defines how the upstream system provides the source video stream to MediaLive. in this tutorial, you create an rtp input.

You must also create an input security group for the input. this input security group applies the rule "only this specific IP address (an IP address that you own) can push to this input on MediaLive." without the protection of this rule, any third party could push content to an MediaLive input if they know the IP address and port of the input.

To create an input and input security group

1. Sign in to the AWS Management Console and open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Inputs**.
3. On the **Inputs** page, choose **Create input**.
4. In the **Input details** section, for **Input name**, enter **`my rtp push`**.
5. For **Input type**, choose **rtp**.

6. in the **Input security group** section, choose **Create**.
7. In the text box, enter the IP address that you noted in the section called "Step 1: Set up the upstream system" of this tutorial. Enter the address as a CIDR block. for example, **`203.0.113.111/32`** and **`203.0.113.112/32`**.
8. Choose **Create input security group**.
9. Choose **Create** to create the input.

   MediaLive adds the input to the list of inputs and automatically creates two destinations (one primary and one redundant). these destinations include the port 5000. for example, **`rtp://198.51.100.10:5000`** and **`rtp://192.0.2.131:5000`**. these are the two locations where the upstream system must push the source.
10. make a note of these two addresses because you will need them in the section called "Step 10: Start the upstream system and the channel".

Step 4: Set up key information

The first step to creating a channel from scratch is to choose the IAM role that MediaLive will use to access the channel when the channel is running (started) and specify key characteristics of the input. Now you are ready to start creating a channel. The first step is to identify the input. The channel contains the details that instruct MediaLive how to transcode (decode and encode) and package that input into specific outputs.

The first step to creating a channel from scratch is to choose the IAM role that MediaLive will use to access the channel when the channel is running (started) and specify key characteristics of the input.

To specify key information for the channel

1. On the MediaLive console, in the navigation pane, choose **Channels**.
2. In the **Channels** section, choose **Create channel**.
3. In the **Channel and input details** pane, in **General info**, for **Channel name**, enter **`Test channel`**.
4. For **IAM role**, choose **Create role from template** and choose **Create IAM role**. The **Use existing role** list now shows the role **`MediaLiveAccessRole`**.
5. Choose **Remember role**.

Step 5: Attach the input

Now you are ready to identify the input that the channel will ingest.

To attach the input to the channel

1. On the **Create channel** page, in the navigation pane, for **Input attachments**, choose **Add**.
2. In **Attach input** , for **Input**, **My RTP push** (the input that you created.)

   The **Attachment name** field is automatically populated with the name of the input itself. You can leave this name as is.
3. Choose **Confirm**. The **Input attachment** section closes, and the **General input settings** section appears.

Step 6: Set up input video, audio, captions

You can create "selectors" to identify the specific video, audio, and captions that you want to extract from the input.

In this tutorial, you don't create a video selector. Instead, when the channel starts, MediaLive will automatically select the video (or the first video) in the input. You also don't create a captions selector Typically, you include captions in the channel configuration, but in this tutorial we omit them.

You do create an audio selector.

To identify the content to extract

1. On the **Create channel** page, in the **Input settings** pane, for **Audio** selectors, choose **Add audio selectors**.
2. For **Audio selector name**, enter **`My audio source`**.

   Ignore the **Selector settings** field. You don't need to specify the PID or language. When the channel starts, MediaLive will automatically select the first audio, which is acceptable for this tutorial.
3. For all other fields in this pane, keep the default values.

Step 7: Create an HLS output group

Once you have set up the input, you continue with the channel creation by creating an output group. In this tutorial, you set up an HLS output group.

To create an output group

1. On the **Create channel** page, in the **Output groups** section, choose **Add**.
2. In the **Add output group** section, choose **HLS**, and then choose **Confirm**.
3. In the **HLS group destination A** section, for **URL**, enter the first input URL that AWS Elemental MediaPackage created for you in the section called "Step 2: Set up the downstream system". For example, **`https://39fuo4.mediapackage.us-east-1.amazonaws.com/in/v1/88dpie/channel`**.
4. For **Credentials**:
   - For **Username**, enter the user name that corresponds to this URL. For example, **`ue739wuty`**.
   - For **Password**, choose **Create parameter**. For **Name**, enter **`DestinationA-MyHLS`**. For **Password**, enter the password that corresponds to the URL. For example, **`due484u`**.
5. Choose **Create parameter**.

   You have created a parameter called **`DestinationA-MyHLS`** that holds the password **`due484u`**. The parameter is stored in the AWS Systems Manager Parameter Store. For more information, see the section called "About the feature for creating password parameters".
6. For **HLS group destination B**, for **URL**, enter the second input URL that AWS Elemental MediaPackage created for you in the section called "Step 2: Set up the downstream system". For example, **`https://mgu654.mediapackage.us-east-1.amazonaws.com/in/v1/xmm9s/channel`**.
7. For **Credentials**:
   - For **Username**, enter the user name that corresponds to this URL. For example, **`883hdux`**.
   - For **Password**, choose **Create parameter**. For **Name**, enter **`DestinationB-MyHLS`**. For **Password**, enter the password that corresponds to the URL. For example, **`634hjik`**.
8. Choose **Create parameter**.

   You have created a parameter called **`DestinationB-MyHLS`** that holds the password **`634hjik`**. The parameter is stored in the AWS Systems Manager Parameter Store.

9. In the **HLS settings** section, for **Name**, enter **MyHLS**.
10. For **CDN settings**, choose **Hls webdav**. This is the connection that AWS Elemental MediaPackage (the downstream system for the channel output) uses.

    Leave the defaults for all the other **CDN settings** fields.
11. For all other fields in this pane, keep the default values.

Step 8: Set up the output and encodes

Now that you have defined one output group in the channel, you can set up an output ins that output group, and specify how you want to encode the video output and the audio output.

To set up the output

1. In the **Output groups** section, choose **Output 1**. MediaLive automatically added this output when you created the output group. In addition, MediaLive automatically set up the output with one video and one audio, as shown in the **Stream settings** section.
2. In **Stream settings**, choose **Video**.
3. For **Video description name**, change the default name to **`H264 video`**.
4. For **Codec settings**, choose **H264**.

   Leave the remaining fields with the default values. Specifically, keep **Width** and **Height** empty to use the same width as the input.
5. In **Stream settings**, choose **Audio 1**.
6. For **Audio description name**, change the default name to **`AAC audio`**.
7. For **Audio selector name**, enter **`My audio source`**, which is the audio selector that you created inthe section called “Step 6: Set up input video, audio, captions”.
8. For **Codec settings**, choose **AAC**.
9. Leave the remaining fields with the default values.

Step 9: Create your channel

You have entered the minimum required information, so you are ready to create the channel.

To create the channel

- On the **Create channel** page, under the **Channel** section, choose **Create channel**.

  The **Channel** section reappears and shows the newly created channel, named **MyHLS**. The state changes to **Creating**, then **Ready**.

Step 10: Start the upstream system and the channel

You can now start the upstream system in order to push the streaming content to MediaLive, encode the content, and send it to AWS Elemental MediaPackage. You can preview the output on MediaPackage.

To start the upstream system

1. In your upstream system, start streaming the video sources that you set up in the section called "Step 1: Set up the upstream system". Set them up to push to the two destinations that you noted in the section called "Step 3: Create an input". These are two addresses in the input in MediaLive. For example, **`rtp://198.51.100.10:5000`** and **`rtp://192.0.2.131:5000`**.
2. On the **Channels** list, choose the channel.
3. Choose **Start**. The channel state changes to **Starting**, then to **Running**.
4. Switch to the web browser tab or window where the AWS Elemental MediaPackage is displayed.
5. Choose the channel link (not the radio button). On the details page, under **Endpoints**, choose **Play**. A preview window appears.
6. Start the video. The output from AWS Elemental MediaLive starts playing.

Step 11: Clean up

To avoid extraneous charges, delete this channel and input when you have finished working with it.

To delete the channel

1. On the **Channels** page, choose the channel.
2. If needed, choose **Stop**.
3. Choose **Delete**.

4. On the **Inputs** page, choose the input.
5. Choose **Delete**.

Setting up AWS Elemental Link

If your organization uses AWS Elemental Link devices with AWS Elemental MediaLive or AWS Elemental MediaConnect, you must deploy the device and configure the device.

AWS Elemental Link (Link) is a *hardware device* that connects a live video source, such as a camera or video production equipment, to MediaLive. The Link device connects to AWS over a secure connection that AWS manages.

Your organization might use Link in one or both of these ways:

- As the video source for the input that you attach to an AWS Elemental MediaLive channel. For more information about this input, see the section called “Elemental Link input”.
- As the video source for an AWS Elemental MediaConnect flow. Only AWS Elemental Link UHD supports this usage. For more information about this input, see Creating a transport stream flow that uses a standard source in the AWS Elemental MediaConnect user guide.

You must perform preliminary setup tasks so that you can use the Link device. Then to use the device, you must configure it to be used in a MediaLive or a MediaConnect workflow.

Topics

HD and UHD devices

There are two versions of the Link device. Each device can handle different usages, ingest different resolutions, and stream different formats.

Device	Usage	Resolutions that the device is ingesting	Resolutions and codecs that the device produces
AWS Elemental Link HD (Link HD)	Connect to a MediaLive input	HD or lower	The same resolution as the ingest, in HEVC
AWS Elemental Link UHD (Link UHD)	Connect to a MediaLive input	UHD or lower	The same resolution as the ingest, in HEVC
	Connect to a MediaConnect flow	UHD or lower	The same resolution as the ingest, in AVC or HEVC

Deploying the AWS Elemental Link hardware

You must deploy the Link device into the AWS cloud.

> **Note**
>
> You don't need to be logged into any AWS service to set up the Link device.

Deploying the hardware

1. To set up the device and the camera that provides the source content, and to connect the device to the internet, see the instructions included in the packaging.

   When you connect the device to the internet, it will verify connectivity by contacting the following domains. If you are watching network traffic, you might see outbound traffic to these domains:

   - amazon.com
   - aws.amazon.com

2. After you have connected the device to the internet, the device automatically connects to MediaLive in the AWS account and the AWS Region that it is configured for.

A user with AWS permissions can view the device on the console, and transfer the device to a different Region.

Performing network diagnostics

If you have problems connecting the device to the internet, you can use the diagnostics utility to troubleshoot these problems.

1. Use the instructions in your Link packaging to connect to the device's on-board user interface.
2. Locate the navigation pane on the left-hand side.
3. Select **Network Diagnostics**, then select **Run diagnostics test** at the top of the page.

   The network diagnostic test starts and takes a few seconds to run.
4. The **Test Information** page appears. This page displays network information and displays the test results: pass or fail (with a reason, and possibly with troubleshooting steps).

The diagnostics feature tests the following:

- **IP address valid** – The configured IP address was successfully applied to the device.
- **Gateway responsive** – There is a connection between the device and the gateway.
- **DNS resolution** – The hostnames resolve to each of the configured DNS servers.
- **AWS connectivity** – There is a connection between the device and AWS over HTTPS.
- **Time server connectivity** – The device can sync the internal time clock using NTP on port 123.
- **Stream connectivity** – The device can send video packets to AWS using port 2088.

Using Link with a MediaLive input

You can set up a Link HD or a Link UHD as the source for an input that you connect to a MediaLive channel. For more information, see the section called “AWS Elemental Link”.

Using Link with a MediaConnect flow

You can set up a Link UHD as the source for a MediaConnect flow.

Topics

- Set up the device in the network
- Set up the Link input device
- Set up the device for the flow
- Monitor the device

Set up the device in the network

Follow this procedure if the Link device is new to your organization.

1. Set up the device on the internet. For more information, see the section called “Deploying the hardware”.
2. Ask your IAM administrator to give you IAM permissions to work with the Link input device interface. See the section called “Setting up users with IAM permissions”.
3. Ask your IAM administrator to set up MediaLive as a trusted entity. See the section called “Setting up MediaLive as a trusted entity”.
4. Sign in to the AWS Management Console and open the MediaLive console at https://console.aws.amazon.com/medialive/.
5. Find out if your organization obtained your device from an AWS reseller. If so, you must claim it.

Set up the Link input device

Use MediaLive to perform these steps.

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Input devices**. The devices that you have access to appear in the **Device list** page. Find the device that you want. If you can't find it, follow the troubleshooting tips in the section called “Viewing details about devices”.
3. Get set up in the correct Region. The device and the flow must be in the same Region. Follow these steps to get aligned:
   - Decide on the Region where you will work.
   - If you want to work in a different Region, transfer the device now. Then switch the MediaLive console to that Region. From now on, make sure that you work in this Region.
4. When the device appears in the **Device list** page, choose the link on the individual card to display the **Device details** page.

5. Check the message on the **Attachments** tab to determine how the device is currently being used.

   Device is not being used

   A message specifies that the device isn't being used, which means that it isn't connected to a MediaLive input or a MediaConnect flow.

   In this case, the device is ready for you to set it up. See the next procedure.

   Device is being used for a flow

   A message specifies that the device is already being used as a source for another MediaConnect flow.

   You can attach a different flow to this device. You should check with other people in your organization to confirm that you can change the usage. You don't have to detach the existing flow, but you might need to stop the device to set it to idle.

   As soon as the device is idle, it's ready for you to set it up. See the next procedure.

   Device is being used for inputs

   A message specifies that the device is already being used as an input source.

   To use this device for a flow, you must first decommission the current usage. You should check with other people in your organization to confirm that no one else plans to use this device is its current usage. Then make a note of all the inputs that this device is attached to. You must delete each input.

   After you delete the last input, the device is ready for you to set it up. See the next procedure.

Set up the device for the flow

Use MediaLive to set up the device.

1. Ask the MediaConnect user in your organization to create a flow. Make sure of the following:
   - The flow must be in the Region that you identified.
   - The flow must use the protocol described as Zixi push for a Link device, and must be set up for a source that is encrypted with AES 128 with a static key. For more information, see the

section about creating a flow with a standard source in the AWS Elemental MediaConnect user guide

2. Obtain the following information from the MediaConnect user:
   - The ARN for the flow.
   - The name of the source for the flow.
   - The ARN for the secret. This secret contains an encryption key. The device will use the encryption key to encrypt the content. MediaConnect must use the same key to decrypt the content it receives.
3. Obtain the following information from your IAM user:
   - The ARN for the role for MediaLive to use to access the flow and the secret. For more information, see the section called "Setting up MediaLive as a trusted entity".
4. Configure the device. If the device was previously used with a different input or flow, review the current configuration and make any necessary changes. For optimum performance, the device must be correctly configured.

   For more information, see the section called "Configuring a device".
5. Choose **Attach MediaConnect flow** or **Edit MediaConnect flow** and specify the new flow. For more information, see the section called "Attaching and detaching a device".

After the flow has become active, you can start the device. (We don't recommend that you start the device before the flow is active.) At the top of the **Device details** tab, choose **Start**. The device starts to stream.

Monitor the device

You can use MediaLive to monitor the device.

- You can view thumbnails of the content, if the device is streaming.
- You can look at metrics to monitor the performance of the device.

Managing Link devices

This section provides reference information for operations you can perform on the Link device. For complete procedures to use Link in a MediaLive channel or a MediaConnect flow, see the section

called “Using Link with a MediaLive input” and the section called “Using Link with a MediaConnect flow”.

Topics

Setting up users with IAM permissions

This section describes the permissions that an IAM administrator must assign to users and other AWS identities so that they can configure a Link device to work with a MediaLive input or an MediaConnect flow.

This information supplements the information about setting up a user to work with all MediaLive features. Read this information as follows:

- Read this section if your organization has users who will only work with MediaLive to deploy devices and configure them for use as sources, and you want to follow a *least permissions* rule.
- If your organization has users who will deploy devices, use those devices, and use all MediaLive features, see the section called “Link”. You should revise their existing policies to include the device permissions.

This section assumes that you have already performed these tasks:

- You have performed the initial setup described in the section called “Preliminary setup steps” in order to sign up for MediaLive and to create an administrator.
- You have read the recommendations in the section called “Identity and Access Management”about how to create administrators, users, and other AWS identities.

Topics

Required permissions

You must assign permissions for actions in several services, as described in the following table.

Permissions	Service name in IAM	Actions

View, configure, and manage a Link device	medialive	`DescribeInputDevice`
		`DescribeInputDeviceThumbnail`
		`ListInputDevices`
		`RebootInputDevice`
		`StartInputDeviceMaintenanceWindow`
		`StartInputDevice`
		`StopInputDevice`
		`UpdateInputDevice`

Handle transfers of Link devices	medialive	`AcceptInputDeviceTransfer` `CancelInputDeviceTransfer`

Permissions	Service name in IAM	Actions

		`ClaimDevice`
		
		`ListInputDeviceTransfers`
		
		`RejectInputDeviceTransfer`
		
		`TransferInputDevice`

On the MediaLive console, view MediaConnect flows in the dropdown list. This dropdown list appears in the **Flow ARN** field in the **Attachments** tab on the **Device details** page.	mediaconnect	`ListFlows`
On the MediaLive console, view Secrets Manager secrets in the dropdown list. This dropdown list appears in the **Secret ARN** field in the **Attachments** tab on the **Device details** page.	secretsmanager	`ListSecrets`
On the MediaLive console, view IAM roles in the dropdown list. This dropdown list appears in the **Role ARN** field in the **Attachments** tab on the **Device details** page.	iam	`ListRoles`

Creating the policy

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane on the left, choose **Policies**. Choose **Create Policy**, then choose the **JSON** tab.
3. In the **Policy editor**, clear the sample content and paste the policy that appears after this procedure.
4. Give the policy a name that makes it clear that this policy is for using Link. For example, `ElementalLinkAccess`.
5. Choose **Create policy**.

Sample policy:

```
{
 "Version": "2012-10-17",
 "Statement": [
   {
     "Effect": "Allow",
     "Action": [
       "medialive:DescribeInputDevice",
       "medialive:DescribeInputDeviceThumbnail",
       "medialive:ListInputDevices",
       "medialive:RebootInputDevice",
       "medialive:StartInputDeviceMaintenanceWindow",
       "medialive:StartInputDevice",
       "medialive:StopInputDevice",
       "medialive:UpdateInputDevice"
     ],
     "Resource": [
       "*"
     ]
   },
   {
     "Effect": "Allow",
     "Action": [
       "medialive:AcceptInputDeviceTransfer",
       "medialive:CancelInputDeviceTransfer",
       "medialive:ClaimDevice",
       "medialive:ListInputDeviceTransfers",
```

```
        "medialive:RejectInputDeviceTransfer",
        "medialive:TransferInputDevice"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "mediaconnect:ListFlows"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:ListSecrets"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:ListRoles"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Setting up MediaLive as a trusted entity

An IAM administrator must consider the special permissions that MediaLive requires if your organization will use a Link device as the source for a MediaConnect flow.

You must set up MediaLive as a *trusted entity*. In a trusted entity relationship, a role identifies MediaLive as a trusted entity. One or more policies are attached to the role. Each policy contains statements about allowed operations and resources. The chain between the trusted entity, role, and policies makes this statement:

"MediaLive is allowed to assume this role in order to perform the operations on the resources that are specified in the policies."

> **Important**
>
> You might be familiar with the trusted entity role that MediaLive needs to work with channels at runtime. We recommend that you create a separate trusted entity role for MediaLive to use with Link devices. The permissions for channels are very complicated. The permissions for devices are very simple. Keep them separate.

Permissions that MediaLive requires

For you to use a Link device, MediaLive must have permissions on operations and resources in MediaConnectand in Secrets Manager:

- For MediaConnect: MediaLive must be able to read details about a flow.
- For Secrets Manager: The device always encrypts the content it sends to MediaConnect. It encrypts using an encryption key that MediaLiveprovides. MediaLive in turn obtains the encryption key from a secret that the MediaConnect user has stored in Secrets Manager. Therefore, MediaLive needs permission to read the the encryption key that is stored in a secret.

This table specifies the required operations and resources.

Permissions	Service name in IAM	Actions	Resources
View the details of a flow	mediaconnect	`DescribeFlow`	All resources
Obtain the encryption key from the secret. See the explanation after this table.	secretsmanager	`GetSecretValue`	The ARN of each secret that holds an encryption key that

Permissions	Service name in IAM	Actions	Resources
			MediaLive needs to access

Topics

Step 1: Create the IAM policy

In this step, you create a policy that makes the statement "Let a principal have access to the specified Secrets Manager actions on the specified resource". Note that the policy doesn't specify a principal. You specify the principal in the next step, when you set up the trusted entity role.

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane on the left, choose **Policies**. Choose **Create Policy**, then choose the **JSON** tab.
3. In the **Policy editor**, clear the sample content and paste the following:

```
{ "Version":  "2012-10-17",
   "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "mediaconnect:DescribeFlow"
        ],
        "Resource": [
          "*"
    ]
  },
     { "Effect":  "Allow",
       "Action": [
         "secretsmanager:GetSecretValue"
      ],
       "Resource": [
      "arn:aws:secretsmanager:Region:account:secret:secret name"
    ]
```

```
    }
  ]
}
```

4. In the **Resource** section for **secretsmanager**, replace the Region, account, and secret name with real values.
5. Add more lines in the **Resources** section or `secretsmanager`, one for each secret. Make sure you include a comma at the end of all lines except the last line. For example:

```
      "Resource": [
        "arn:aws:secretsmanager:us-west-2:111122223333:secret:emx_special_skating-
KM19jL",
        "arn:aws:secretsmanager:us-
west-2:111122223333:secret:aes-":secret:emx_weekly_live_poetry-3ASA30",
        "arn:aws:secretsmanager:us-
west-2:111122223333:secret:aes-":secret:emx_tuesday_night_curling-AMcb01"
      ]
```

6. Give the policy a name that makes it clear that this policy is for Link and a flow. For example, `medialiveForLinkFlowAccess`.
7. Choose **Create policy**.

Step 2: Set up the trusted entity role

In this step, you create a role that consists of a trust policy ("let MediaLive call the `AssumeRole` action") and a policy (the policy that you just created). In this way, MediaLive has permission to assume the role. When it assumes the role, it acquires the permissions specified in the policy.

1. On the IAM console, in the navigation pane on the left, choose **Roles**, then **Create Role**. The **Create role** wizard appears. This wizard walks you through the steps of setting up a trusted entity, and adding permissions (by adding a policy).
2. On the **Select trusted entity** page, choose the **Custom trust policy** card. The **Custom trust policy** section appears, with a sample policy.
3. Erase the sample, copy the following text, and paste the text in the **Custom trust policy** section. The **Custom trust policy** section now looks like this:

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
 {
            "Effect": "Allow",
            "Principal": {
                "Service": "medialive.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

4. Choose **Next**.
5. On the **Add Permissions** page, find the policy that you created (for example, `medialiveForLinkFlowAccess`), and select the checkbox. Then choose **Next**.
6. On the review page, enter a name for the role. For example, `medialiveRoleForLinkFlowAccess`.
7. Choose **Create role**.

Claiming a Link device

If you purchase a device from an AWS reseller, you must claim the device.

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the MediaLive console, look for the Region menu in the navigation bar at the top of the page. Switch to the us-west-2 Region.
3. In the navigation pane, choose **Input devices**. The **Claim a device** button appears. (This button only appears in this Region.)
4. Choose **Claim a device** and enter the ID of the device. For example, **hd-0000aaaa1111bbbb2222cccc** or **uhd-9999aaaa8888bbbb7777cccc**.

   The device now appears in the list of devices.
5. Find the card for the Link input device. If there are many Link input devices, enter part of the name to filter the list.

Creating a Link input device

Within MediaLive, the Link device is represented a resource that is called theLink input device. You don't need to create this resource. Instead, when an user connects a Link device to the internet and

powers it on, the device automatically connects to MediaLive in your AWS account, and specifically to MediaLive in the AWS Region that it is configured for.

Viewing your Link devices

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Input devices**. The devices that you have access to appear in the **Device list** page.
3. If you don't see **Link input device** listed in the navigation pane, or if you don't see a card for the device that you want, you are probably in the wrong Region, follow the troubleshooting tips after this procedure.
4. Find the card for the Link input device that you want. If there are many Link input devices, enter part of the name to filter the list.
5. Choose the hyperlink. The **Device details** page appears. The page consists of an **Aggregate status** panel, and a panel with three tabs.

Troubleshooting tips

If you can't find an expected device in the list, try the following:

- Another user in your organization might have moved the device to another Region.

  Determine the Region that the device is in. In the MediaLive console, look for the Region menu in the navigation bar at the top of the page. Switch to the appropriate Region.
- This might be a new device that you purchased from AWS. In this case, the device initially appears in the Region you specified when you purchased the device.

  In the MediaLive console, look for the Region menu in the navigation bar at the top of the page. Switch to the appropriate Region.
- This might be a new device that you purchased from an AWS reseller. In this case, the device is in the us-west-2 Region.

  In the MediaLive console, look for the Region menu in the navigation bar at the top of the page. Switch to the us-west-2 Region. If the device still doesn't appear, you probably need to claim the device.

Topics

Aggregate status panel

The panel includes the sections and fields specified in this table.

Section	Fields	Details
Title line	The unique ID of the Link hardware	
	Synchronization state	
	Software status	To update software, see the section called “Updating device software”.
Aggregate status	A device thumbnail	A thumbnail of the content that is currently being pushed by the device, if there is any being pushed. The device generates the thumbnails by capturing a video frame approximately every 5 seconds.
	The device state, connection state, video resolution (WxH), active input	Only shows information if the device is connected to AWS and is sending content.

Details tab

This tab includes the sections and fields specified in this table.

Section	Fields	Details
Network settings	The current network settings	Shows information only if the device is connected to AWS. To connect, see the section called “Deploying the hardware”.
Device settings	The current configuration of the device.	The streaming fields (such as the framerate) show information only if the device is sending content to AWS. You can set some fields. Choose the **Modify** then **Configure device**. For information about the dialog, see the section called “Configuring a device”.
Device metadata	The Device ARN (which includes the unique device ID), serial number, device ID, type, name of device	

Attachments tab

This tab displays information about how the device is currently being used — whether the device is connected to a MediaLive input, a MediaConnect flow, or isn't being used.

You can change the flow that the device is attached to. You can detach the flow, so that the device isn't in use. For more information about both these tasks, see the section called “Attaching and detaching a device”.

Device is not configured for any use

A message indicates that the device is not being used. To set up the device as the source for an input, see the section called “Using Link with a MediaLive input”. To set up the device as the

source for a MediaConnect flow (UHD devices only), see the section called “Using Link with a MediaConnect flow”.

Device is configured as the source for one or more inputs

The tab displays the ARN of the MediaLive input that the device is attached to. You can set up the device as the source for up to four inputs. You can detach the input so that the device isn't in use. For more information about both these tasks, see the section called “Attaching and detaching a device”.

Device is configured as the source for an MediaConnect flow

Applies only to UHD devices. The tab displays information about the attachment to the flow:

- The ARN of the flow
- The source name of the flow. Keep in mind that flows can have more than one source.
- The ARN of the secret used for encryption.
- The ARN of the role that MediaLive uses to work with the flow.

Tags tab

This table displays tags that you have set up for the device. You can choose **Add tag** and **Remove tag** to change tags. For more information about tags, see Tagging AWS Elemental MediaLive resources.

Transferring a Link device to another account

You can transfer a device to a different AWS account, to transfer ownership of the device to that account. The recipient of a transfer must accept or reject the incoming transfer in order for the transfer to be finalized. After the transfer is finalized, all charges for use of the device are applied to the new account.

Topics

- Initiating a device transfer

Initiating a device transfer

To transfer a Link to another AWS account

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Input devices**. Find the card for the Link that you want to transfer and choose the hyperlink.
3. On the **Device details** page for the device, choose **Other device actions** then **Transfer device**.
4. On the **Transfer input device** dialog, choose **Transfer to another AWS account**, enter the AWS account to transfer to, and type an optional message. Then choose **Transfer**.
5. In the navigation pane, choose **Input devices**, then choose **Device transfers**. The transfer request appears in the **Outgoing transfers** tab.

   The transfer is pending until the recipient accepts the device. While the transfer is pending, you can cancel the request, as described in the following section.

   If the recipient accepts the transfer, the device no longer appears in any of your device lists.

   If the recipient rejects the transfer, the device appears again on your **Input devices** page.

Cancelling an outgoing device transfer

You can cancel a device transfer while the request is pending.

To cancel an outgoing device transfer

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Input Devices**. Choose **Device transfers**, then choose the **Outgoing transfers** tab.
3. In the list of transfers, choose the transfer you want to cancel, then choose **Cancel**.

Accepting a device transfer

The owner of a device can transfer a device to your AWS account. For example, someone in your organization might transfer the device from one AWS account in your organization to another AWS account.

If you are expecting to receive a device transfer, you should regularly check the **Incoming transfers** tab on the **Device transfers** page. You must accept the transfer. You can't use the device until you have accepted the transfer.

To accept a device transfer

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Input devices**. Choose **Device transfers**, then choose the **Incoming transfers** tab.
3. In the list of transfers, choose the device that you want to accept, then choose **Accept** or **Reject**.
4. In the navigation pane, choose **Input devices** again. The device now appears in the list of devices on the **Input devices** page.

Transferring a Link device to another Region

You can transfer a device to a different AWS Region. (If instead you want to transfer the device to a different Availability Zone in the existing Region, see the section called “Configuring a device”.)

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Input devices**. Find the card for the Link that you want to transfer, and choose the hyperlink.
3. On the **Device details** page for the device, choose **Other device actions** then **Transfer device**.
4. On the **Transfer input device** dialog, choose **Transfer to another AWS Region**, and enter the Region. Then choose **Transfer**. The transfer occurs immediately. There is no need to confirm the transfer.
5. To locate the moved device, switch to the target Region. Choose **Input devices** in the left navigation pane. The devices that you have access to appear.

   You should review the configuration because any customizations (such as the Availability Zone) are deleted during the transfer.

Configuring a Link device

The Link device has properties that control how it behaves when it is streaming. You can set these properties.

Guidelines for changing configuration fields

You can change most fields at any time, including when the device is attached to an active flow or a running channel.

Changing the Availability Zone

You can change the Availability Zone only when the device is attached to a MediaLive input. Typically you change the Availability Zone only when you're deploying it to use it with a new channel. Even if you stop the channel (which automatically stops the device), you can't simply change the Availability Zone. Doing so would probably make the channel inoperable.

Changing the codec for an input

There is no point changing the codec when a device is attached to a MediaLive input. Only one codec (HEVC) is supported.

Changing the codec for a flow

You might want to change the codec when a UHD device is attached to a MediaConnect flow. (Only UHD devices offer a choice of codecs.) To do this, you must first stop the device. Stopping the device has no impact on the flow. The flow remains active, even though it isn't receiving content. Make then change, then restart the device. The flow starts to receive content again.

Managing the codec in multi-use scenarios

You might switch between using a UHD device for a MediaLive input and a MediaConnect flow. But these two usages support different codecs with that device.

- A MediaLive input only supports AVC with a UHD device.
- A MediaConnect flow only supports HEVC with a UHD device.

Consider this scenario: you use a UHD device with a MediaConnect flow and set the codec to AVC. You later attach the device to a MediaLive input. When the device starts (and it's connected to

an input), MediaLive will automatically change the codec to HEVC in the configuration, and will encode using HEVC.

When you switch back to using the device with a flow, you should check the codec, because it might not be what you expect.

To configure a device

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Link input devices**. Find the Link that you want. If there are many devices listed, enter part of the name to filter the list.
3. Choose the hyperlink for the device. The **Device details** page for this device appears.
4. Choose **Modify** then **Configure device**. Change fields. For details about each field, see the sections after this procedure. Then choose **Update**.

   MediaLive sends the new values (except for the name) to the devices, so that the device can update itself.
5. Display the **Device details** page and take the appropriate action:
   - If the device is not attached to a MediaLive input or a MediaConnect flow, watch for the state of the device to return to **Idle**. You can now start the channel (which automatically starts the device), or you can start the device and the flow.
   - If the device is currently attached to a MediaLive input or a MediaConnect flow that is active, watch for the state of the device to return to **In use**. Note that there is no need for you to restart the input or flow.

Configuration fields

Field	Description
Name	Enter a descriptive nickname for the device, for use only in MediaLive. We recommend that you give the name a prefix (such as **hd-**) to indicate that this is a Link HD.
Input source	Configures the source that Link sends to MediaLive: SDI or HDMI. If the device usually has only one source, choose **Auto**. Link will send the only source, or will send the first source that it encounters.

Field	Description
	If the device frequently has sources active on both SDI and HDMI, you must specify which source you want the device to send. You must change this value whenever you want to use the other source.
Maximum bitrate	Set a value if you want to throttle the delivery bitrate to MediaLive. Leave this field blank to let the device determine the bitrate that is best for the network conditions between the device and MediaLive.
Latency	Set the device buffer size (latency). A higher latency value means a longer delay in transmitting from the device to MediaLive, but improved resiliency. A lower latency value means a shorter delay, but less resiliency. If you leave this field blank, the service uses the default value. Default: 1000 ms (HD devices) or 2000 ms (UHD devices). Minimum: 0 ms. Maximum: 6000 ms.
Availability Zone	The Availability Zone (AZ) for the device. This field has an effect only when the device is attached to a MediaLive input. It is ignored when the device is attached to a flow. You should change the Availability Zone for the device whenever the device needs to belong to a different Availability Zone. Key use cases for changing the Availability Zone are the following: • If you want to use the device in a single-pipeline channel with multiple inputs, where several inputs are *Availability Zone aware*. All the inputs must use the same Availability Zone (or the two Availability Zones, for a standard-class input). • If you want to use the device in a standard channel (two pipelines) in order to implement pipeline redundancy.

Field	Description
Codec	This field appears only if the device is a UHD device. Choose the codec for the device to use to encode the video that it outputs. The default is HEVC. If you set this field with a codec that the device usage doesn't support, MediaLive will reset the value to the default codec. Typically, this behavior occurs when you deploy the device in a compliant way and then switch to a deployment where the specified codec isn't supported.

Attaching and detaching a Link device

If you are using a Link device as the source for a MediaConnect flow, you must attach the flow to the device. You can also detach the flow to stop using the device as the source for that flow.

To attach a device to a flow

1. Obtain the information about the flow from the person who created the flow.
2. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
3. In the navigation pane, choose **Input devices**. Find the card for the appropriate Link , and choose the hyperlink.
4. Choose **Attach MediaConnect flow**. Complete the fields:

   - Flow ARN: The ARN of the flow that you obtained from the MediaConnect user. Either choose **List of ARNs** and select the ARN, or choose **Manual input** and type the ARN.

   - Source name: Type the name that you obtained from the MediaConnect user. Keep in mind that the flow might an have more than one source, so make sure that you obtain the correct name.
   - Secret ARN: The ARN of the secret that holds the encryption key to use with this flow. You obtained this value from the MediaConnect user.
   - Role ARN: The ARN of the role that MediaLive must assume. Obtain this value from your IAM administrator.
5. Choose **Save**.

The device is attached to the specified flow. When you later start the device, MediaLive uses the role ARN to obtain the encryption key that is stored in the secret. MediaLive delivers the key to the device, and the device encrypts the content that it streams. The MediaConnect flow uses the same key to decrypt the content as it receives it.

To detach a flow from device

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Input devices**. Find the card for the appropriate Link , and choose the hyperlink.

   If the **Device details** page displays information about a MediaConnect flow, then you know that the device is currently attached to a flow.
3. Choose **Remove MediaConnect flow**. Then choose **Save**.

To detach a device from an input

To remove the connection between a device an Elemental Link input, you can do either of these changes:

- You can ??? so that it is connected to a different device.
- You can delete the input. Note that you can't modify an Elemental Link input so that it doesn't have a device connected to it.

Starting and stopping a Link device

You must start or stop a Link only if it is configured as the source for a MediaConnect flow. (You don't need to start or stop the device when it is configured as the source for a MediaLive input. In this case, MediaLive automatically starts and stops the device when you start and stop the related channel.)

You must start a device to instruct it to start streaming video content to send to MediaConnect. After you start a device, it will always try to stream content. It will only stop trying when you explicitly stop it. This means, for example, that if you reboot a device, streaming will automatically resume after the reboot.

You must stop a device before you can perform the following actions:

- Update some settings
- Attach the flow to a different device or detach the flow from the device.

To start or stop a device

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Input devices**. Find the card for the appropriate Link , and choose the hyperlink.
3. On the **Device details** page for the device, choose **Start** or **Stop**.

Rebooting a Link device

You can remotely reboot a Link device from the AWS console. You don't need physical access to the device.

Typically, you reboot the device only as a last resort, to resolve a problem with the device response or with the content that is streaming.

You don't have to stop either the device or the channels before rebooting. MediaLive handles the reboot smoothly.

To reboot a device

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Input devices**. Find the card for the Link that you want to reboot, and choose the hyperlink.
3. On the **Device details** page for the device, choose **Other device actions** then **Reboot device**, then confirm the request.

   Any MediaLive channels or MediaConnect flows that are using the device will lose input briefly, but the channels or flows won't stop or fail.

   When the reboot is complete, the device connection status changes to **Connected**. If the device was streaming prior to the reboot, it automatically resumes streaming.

Updating software on a Link device

Link devices automatically install updates when they are powered on, assuming that the MediaLive channels that use the device have stopped.

However, if you don't stop your channels frequently, you should start a maintenance window at a convenient time. The device will install software updates some time in the next two hours.

You don't have to stop either the device or the channels before starting a maintenance window. MediaLive handles the update smoothly.

To start a maintenance window for a device

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Input devices**. Find the card for the Link that you want to update, and choose the hyperlink.
3. On the **Device details** page for the device, choose **Other device actions** then **Start maintenance**, then confirm the request.

   The maintenance window starts. The update will start some time during the next two hours. When the update starts, any channels that are using the device will lose input briefly, but the channels won't stop or fail.

   When the reboot is complete, the device connection status changes to **Connected**. If the device was streaming prior to the reboot, it automatically resumes streaming output.

Deleting a Link input device

You don't delete Link input devices. Instead, if someone deregisters the Link device, the Link input device (which is the interface for the device in the console) no longer appears in the **Devices** section. Note that this is the only way that the Link input device is ever removed.

- If someone powers down the device, the Link input device still appears in the list.
- If the device is disconnected from the internet, or if the connection from MediaLive to the Link device is down, the Link input device still appears in the list.

Preparing the upstream and downstream systems in the MediaLive workflow

From the point of view of AWS Elemental MediaLive, a live streaming workflow that includes MediaLive involves three systems:

- An *upstream system* that provides the video content to MediaLive.
- MediaLive, which ingests the content and transcodes the content.
- A *downstream system* that is the destination for the output that MediaLive produces.

You should plan that workflow before you start to create the channel. As the first stage in that planning, you must set up the upstream and downstream systems. As the second stage, you must plan the channel itself—identify the content to extract from the source content, and plan the outputs to produce.

This chapter deals with preparing the upstream and downstream sections. *Setup: Planning the channel* deals with planning the channel.

> **Important**
>
> This procedure describes planning the workflow starting from the output and then working back to the input. This is the most effective way to plan a workflow.

To plan the MediaLive workflow

1. Identify the output groups that you need to produce, based on the systems that are downstream of MediaLive. See the section called “Step 1: Identify output group types”.
2. Identify the requirements for the video and audio encodes that you will include in each output group. See the section called “Step 2: Identify encode requirements”.
3. Decide on the channel class—decide if you want to create a standard channel that supports redundancy or a single-pipeline channel that doesn't support redundancy. See the section called “Step 3: Identify resiliency requirements”.
4. Assess the source content to make sure it's compatible with MediaLive and with the outputs that you need to create. For example, make sure that the source content has a video codec that MediaLive supports. See the section called “Step 4: Assess the upstream system ”.

After you have performed these four steps, you know whether MediaLive can handle your transcoding request.

5. Collect identifiers for the source content. For example, ask the operator at the upstream system for the identifiers for the different audio languages that you want to extract from the content. See the section called “Step 5: Collect information about the source content”.
6. Coordinate with the downstream system or systems to provide a destination for the output groups that MediaLive will produce. See the section called “Step 6: Coordinate with downstream systems”.

Topics

Step 1: Identify the output group types for the downstream system

The first step in planning any AWS Elemental MediaLive workflow is to determine which types of *output groups* you need to produce, based on the requirements and capabilities of the systems that are downstream of MediaLive.

Perform this work with the downstream system before you assess the upstream system. Decision making in a workflow starts with the downstream system, then works back to the upstream system.

Important

You should have already identified the downstream system or systems that you are going to send MediaLive output to, for this workflow. If you have not yet identified the

downstream system, you must do some research before continuing with preparing your workflow. This guide can't help you to identify your downstream system. When you know what your downstream systems are, return to this section.

To identify the output group

1. Obtain the following information from your downstream system.
   - The required output formats. For example, HLS.
   - The application protocol for each. For example, HTTP.
2. Decide on the delivery mode for your outputs.
   - You might have an output that is on a server that is on your EC2 instance in your VPC. Or you might have an output that is in Amazon S3. If one or both of these situations apply, you might want to set up for delivery via your VPC. For more information, see the section called "VPC delivery".
   - If you don't have any of these types of outputs, you will deliver in the regular way.
3. Make sure that MediaLive includes an *output group* that supports the output format and protocol that the downstream system requires. See the section called "Supported output types".
4. If your preferred downstream system is another AWS media service, read this for information about choosing the service.
5. If your downstream system supports Microsoft Smooth Streaming, see the section called "Options for Microsoft Smooth" for options.
6. Decide if you want to create an archive output group in order to produce an archive file of the content. An archive file is a supplement to streaming; it isn't itself a streaming output. Typically, you create an archive file as a permanent file version of the streaming output.
7. Decide if you want to create a frame capture output group in order to produce a frame capture output. A frame capture output is a supplement to streaming; it isn't itself a streaming output. This type of output might be useful for your workflow. For example, you might use a frame capture output to create thumbnails of the content.
8. Make a note of the output groups that you decide on. You will need this information when you design the output groups.

   For example, after you have followed these steps, you might have this list of output groups:

- One HLS output group with AWS Elemental MediaPackage as the downstream system.
- One RTMP output group sending to the downstream system of a social media site.
- One Archive output group as a record.

Topics

Choosing among the AWS media services

If your preferred downstream system is another AWS media service, following are some useful tips for choosing the service to use:

- If you need to choose between AWS Elemental MediaPackage or AWS Elemental MediaStore for HLS outputs, follow these guidelines:
  - Decide if you want to protect your content with a digital rights management (DRM) solution. DRM prevents unauthorized people from accessing the content.
  - Decide if you want to insert ads in your content.

  If you want either or both of these features, you should choose MediaPackage as the origin service because you will need to repackage the output.

  If you do not want any of these features, you could choose MediaPackage or AWS Elemental MediaStore. AWS Elemental MediaStore is generally a simpler solution as an origin service, but it lacks the repackaging features of MediaPackage.
- If you have identified AWS Elemental MediaPackage as an origin service, decide if you will produce the HLS output using an HLS output group or a MediaPackage output group. For guidelines on making this choice, see the next section.

Choosing between the HLS output group and MediaPackage output group

If you want to deliver HLS output to AWS Elemental MediaPackage, you must decide if you want to create an HLS output group or a MediaPackage output group.

Delivering to MediaPackage v2

If you are delivering to a MediaPackage channel that uses MediaPackage v2, you must create an HLS output group. The MediaPackage operator can tell you if the channel uses version 2 of the API. One use case for using version 2 is to implement a glass-to-glass low latency workflow that includes both MediaLive and MediaPackage.

Delivering to standard MediaPackage (v1)

There are differences in the setup of each type of output group:

- The MediaPackage output requires less setup. AWS Elemental MediaLive is already set up with most of the information that it needs to package and deliver the output to the AWS Elemental MediaPackage channel that you specify. This easier setup has benefits, but it also has drawbacks because you can't control some configuration. For information about how MediaLive sets up a MediaPackage output group, see the section called “Result of this procedure”.
- For a MediaPackage output, the MediaLive channel and the AWS Elemental MediaPackage channel must be in the same AWS Region.
- In a MediaPackage output, there are some restrictions on setting up ID3 metadata. For details, see the section called “ID3 metadata”.

Options for handling Microsoft Smooth output

If you are delivering to a Microsoft Smooth Streaming server, the setup depends on whether you want to protect your content with a digital rights management (DRM) solution. DRM prevents unauthorized people from accessing the content.

- If you don't want to implement DRM, then create a Microsoft Smooth output group.
- If you do want to implement DRM, you can create an HLS or MediaPackage output group to send the output to AWS Elemental MediaPackage, then use AWS Elemental MediaPackage to add DRM. You will then set up AWS Elemental MediaPackage to deliver to the Microsoft Smooth origin server.

Step 2: Identify the encode requirements for the output groups

After you have identified the output groups that you need to create, you must identify the requirements for the video and audio encodes that you will include in each output group. The downstream system controls these requirements.

Perform this work with the downstream system before you assess the upstream system. Decision making in a workflow starts with the downstream system, then works back to the upstream system.

To identify the video and audio codecs in each output group

Perform this procedure on every output group that you identified.

1. Obtain the following video information from your downstream system:
   - The video codec or codecs that they support.
   - The maximum bitrate and maximum resolution that they can support.
2. Obtain the following audio information from your downstream system:
   - The supported audio codec or codecs.
   - The supported audio coding modes (for example, 2.0) in each codec.
   - The maximum supported bitrate for audio.
   - For an HLS or Microsoft Smooth output format, whether the downstream system requires that the audio is bundled in with the video or that each audio appears in its own rendition. You will need this information when you organize the assets in the MediaLive outputs.
3. Obtain the following captions information from your downstream system.
   - The captions formats that they support.
4. Verify the video. Compare the video codecs that your downstream system requires to the video codecs that MediaLive supports for this output group. See the tables in the section called "Supported codecs by output type". Make sure that at least one of the downstream system's offered codecs is supported.
5. Verify the audio. Compare the audio codecs that your downstream system requires to the video codecs that MediaLive supports for this output group. See the tables in the section called "Supported codecs by output type". Make sure that at least one of the downstream system's offered codecs is supported.

6. Skip assessment of the caption formats for now. You will assess those requirements in a later section.
7. Make a note of the video codecs and audio codecs that you can produce for each output group.
8. Decide whether you want to implement a trick-play track. For more information, see the section called "Trick-play track".

Result of this step

After you have performed this procedure, you will know what output groups you will create, and you will know which video and audio codecs those output groups can support. Therefore, you should have output information that looks like this example.

Example

Output group	Downstream system	Video codecs supported by downstream system	Audio codecs supported by downstream system
HLS	MediaPackage	AVC	AAC 2.0, Dolby Digital Plus
RTMP	social media site	AVC	AAC 2.0
Archive	Amazon S3	The downstream system doesn't dictate the codec—you choose the codec that you want.	The downstream system doesn't dictate the codec—you choose the codec that you want.

Step 3: Identify resiliency requirements

Resiliency is the ability of the channel to continue to work when problems occur. MediaLive includes two resiliency features that you must plan for now. You must decide which of these features you want to implement. You must make this decision now because these features affect how many sources you need for your content, which requires discussion with your upstream system.

Pipeline redundancy

You can set up a channel with two pipelines, to provide resiliency within the channel processing pipeline.

Pipeline redundancy is a feature that applies to the entire channel and to all the inputs attached to the channel. Early on in your planning of the channel, you must decide how you want to set up the pipelines.

You set up for pipeline redundancy by setting up the channel as a *standard channel* so that it has two encoding pipelines. Both pipelines ingest the source content and produce output. If the current pipeline fails, the downstream system can detect that it is no longer receiving content and can switch to the other output. There is no disruption to the downstream system. MediaLive restarts the second pipeline within a few minutes.

For more information on pipeline redundancy, see the section called “Pipeline redundancy”.

Automatic input failover

You can set up two push inputs for automatic input failover, to provide resiliency for one input in the channel.

Automatic input failover is a feature that applies to individual inputs. You don't have to make a decision about implementing automatic input failover when planning the channel. You can implement it later on, when attaching a new push input, or when you want to upgrade an existing push input so that it implements automatic input failover.

To set up for automatic input failover, you set up two push inputs (that have the exact same source content) as an *input failover pair*. Setting up this way provides resiliency in case of a failure in the upstream system, or between the upstream system and the channel.

In the input pair, one of the inputs is the *active* input and one is on *standby*. MediaLive ingests both inputs, in order to always be ready to switch, but it usually discards the standby input immediately. If the active input fails, MediaLive immediately fails over and starts processing from the standby input, instead of discarding it.

You can implement automatic input failover in a channel that is set up for pipeline redundancy (a standard channel) or one that has no pipeline redundancy (a single-pipeline channel).

For more information about automatic input failover, see the section called “Automatic input failover”.

Comparison of the two features

Following is a comparison of pipeline redundancy and automatic input failover.

- There is a difference in the failure that each feature deals with:

  Pipeline redundancy provides resiliency in case of a failure in the MediaLive encoder pipeline.

  Automatic input failover provides resiliency in case of a failure ahead of MediaLive, either in the upstream system or in the network connection between the upstream system and the MediaLive input.
- Both features require two instances of the content source, so in both cases your upstream system must be able to provide two instances.

  With pipeline redundancy, the two sources can originate from the same encoder.

  With automatic input failover, the sources must originate from different encoders, otherwise both sources will fail at the same time, and the input failover switch will fail.
- Pipeline redundancy applies to the entire channel. Therefore you should decide whether you want to implement it when you plan the channel. Automatic input failover applies to only one input. Therefore you could, for example, decide to implement automatic input failover only when you attach your most important push input.
- Automatic input failover requires that the downstream system be able to handle two instances of the output and be able to switch from one (when it fails) to the other. MediaPackage, for example, can handle two instances.

  If your downstream system doesn't have this logic built in, then you can't implement automatic input failover.

Step 4: Assess the upstream system

As part of the planning of the MediaLive workflow, you must assess the upstream system that is the source of the content, to ensure that it is compatible with MediaLive. Then you must assess the source content to ensure that it contains formats that MediaLive can ingest and that MediaLive can include in the outputs you want.

You obtain the *source content* from a *content provider*. The source content is provided to you from an *upstream system* that the content provider controls. Typically, you have already identified the

content provider. For more information about source content and upstream systems, see *How MediaLive Works*.

To assess the upstream system

1. Speak to the content provider to obtain information about the upstream system. You use this information to assess the ability of MediaLive to connect to the upstream system, and to assess the ability of MediaLive to use the source content from that upstream system.

   For details about the information to obtain and assess, see the following sections:

   - the section called "Assess source formats and packaging"
   - the section called "Assess video content"
   - the section called "Assess audio content"
   - the section called "Assess captions"
2. Make a note of the MediaLive input type that you identify for the source content.
3. Make a note of the following three characteristics of the source stream. You will need this information when you set up the channel:

   - The video codec
   - The resolution of the video—SD, HD, or UHD
   - The maximum input bitrate

Result of this step

At the end of this step, you will be confident that MediaLive can ingest the content. In addition you will have identified the following:

- The type of MediaLive input you will create to ingest the source content.
- The information that you need to extract the video, audio, and captions from the source (from the MediaLive input). For example:

Information	Format	Characteristics
Source formats and packaging	RTP	with FEC

Information	Format	Characteristics
Supported video codecs	HEVC	1920x1080 5 Mbps maximum
Supported audio codecs, coding modes, and languages	Dolby Digital 5.1	English, Spanish
	AAC 2.0	English, Spanish, French, German
Supported captions formats	Embedded	English, Spanish, French, German
	Teletext	10 languages

Assess source formats and packaging

Consult the following table for information about how to assess the source formats and packaging. Read across each row.

Information to obtain	Verify the following
Number of sources that the content provider can provide.	If you plan to implement a resiliency feature, make sure that your content provider can deliver the required inputs: • For automatic input failover, they must deliver two identical instances of the same source content. • For pipeline redundancy, they must deliver two identical instances of the same source content. • If you plan to implement both features, they must deliver four instances.
Delivery formats and protocols	Find out what format and protocol the upstream system supports for delivery.

Information to obtain	Verify the following
The type of MediaLive input that applies to the format you identify	Make sure that this format is listed in the table in the section called "Input types, protocols, upstream systems". • If the format is listed, identify the MediaLive input type that applies to the format. • If the format isn't listed, speak to your content provider about how they can add support for MediaLive. Note that you don't need to verify this information for content delivered over CDI or content delivered from an AWS Elemental Link. MediaLive can always handle these input types.
Whether the upstream system is using the latest SDK	Make sure that the content provider is using the latest version of the AWS CDI SDK on their upstream CDI source device.
Whether the source content is a stream or VOD asset	Find out if the source content is a live stream or a VOD asset. Make sure that MediaLive supports the delivery for the format that you identified. See the table in the section called "Support for live and file sources".

Information to obtain	Verify the following
Whether the content is encrypted	MediaLive can ingest encrypted content only from HLS content. If the source content is HLS and it is encrypted , make sure that it is encrypted in a format that MediaLive supports. See the section called “Encrypted HLS content”. If MediaLive doesn't support the available encryption format, find out if you can obtain the content in unencrypted form.
Only if the source content is RTP, whether it includes FEC.	We recommend that the source content include FEC because it is less likely to result in an output that has visual disruptions.

Handling encrypted source content in an HLS source

MediaLive can ingest an HLS source that is encrypted according to the HTTP Live Streaming specification.

Supported encryption format

MediaLive supports the following format for encrypted HLS sources:

- The source content is encrypted with AES-128. MediaLive doesn't support AES-SAMPLE.
- The source content is encrypted using either static or rotating keys.
- The manifest includes the `#EXT-X-KEY` tag with these attributes:
  - The `METHOD` attribute specifies AES-128.
  - The URI specifies the license server for the encryption key.
  - The IV is blank or specifies the initialization vector (IV) to use. If the IV is blank, MediaLive uses the value in the `#EXT-X-MEDIA-SEQUENCE` tag as the IV.
- If both the upstream system and the license server require authentication credentials (user name and password), make sure that the same credentials are used on both servers. MediaLive does not support having different credentials for these two servers.

How decryption works

The content owner sets up the main manifest to include the `#EXT-X-KEY` with the method (AES-128), the URL to the license server, and the initialization vector (IV). The content owner places the encryption keys on the license server. When the MediaLive channel that uses this source starts, MediaLive obtains the main manifest and reads the `#EXT-X-KEY` tag for the URL of the license server.

MediaLive connects to the license server and obtains the encryption key. MediaLive starts pulling the content from the upstream system, and decrypts the content using the encryption key and the IV.

Assess video content

Consult the following table for information about how to assess video source. Read across each row.

> **Note**
>
> You don't need to perform any assessment of the video being delivered over CDI or from an AWS Elemental Link device. These sources are always acceptable to MediaLive.

Information to obtain	Verify the following
The available video codecs or formats.	Make sure that at least one of the video codecs is included in the list of video codecs for the package format. See the section called "Input codecs". If the content is available in more than one supported codec, decide which single video codec you want to use. You can extract only one video asset from the source content.
The available resolutions.	MediaLive supports only landscape mode.
The maximum expected bitrate.	Make sure that the bandwidth between the upstream system and MediaLive is sufficient

Information to obtain	Verify the following
	to handle the anticipated maximum bitrate of the source content. If you are setting up standard channels (to implement pipeline redundancy), make sure that the bandwidth is double the anticipat ed maximum bitrate because there are two pipelines.
Whether the video characteristics change in the middle of the stream.	For best results, verify that the video character istics of the video source don't change in the middle of the stream. For example, the codec should not change. The frame rate should not change.

Assess audio content

Consult the following table for information about how to assess the audio source. Read across each row.

> **Note**
>
> You don't need to perform any assessment of the audio being delivered over CDI or from an AWS Elemental Link device. These sources are always acceptable to MediaLive.

Information to obtain	Verify the following
The available audio codecs or formats.	Make sure that at least one of the audio codecs is included in the list of audio codecs in the section called “Input codecs”.

Information to obtain	Verify the following
The available languages for each codec. For example, English, French.	Identify the languages that you would like to offer. Determine which of these languages the content provider can provide.
The available coding modes (for example, 2.0 and 5.1) for each codec.	Identify the audio coding modes that you prefer for each audio language. Determine which of these coding modes the content provider can provide. For more information, see the section after this table.
Whether the audio characteristics change in the middle of the stream.	For best results, verify that the audio characteristics of the source content don't change in the middle of the stream. For example, the codec of the source should not change. The coding mode should not change. A language should not disappear.
If the source content is HLS, whether the audio assets are in an audio rendition group or multiplexed with video.	MediaLive can ingest audio assets that are in a separate rendition group or multiplexed into a single stream with the video.

To decide on a coding mode

If multiple coding modes are available for the same language, decide which mode you want to use. Follow these guidelines:

- You can extract some languages in one codec and coding mode, and other languages in another codec and coding mode. For example, you might want one or two languages available in 5.1 coding mode, and want other languages in 2.0 coding mode.

- You can extract the same language more than once. For example, you might want one language in both 5.1 coding mode and coding mode 2.0.
- When deciding which codec and coding mode to extract for a given language, consider the coding mode you want for that language in the output. For each language, it is always easiest if

the coding mode of the source content matches the coding mode of the output, because then you don't have to remix the audio in order to convert the coding mode. MediaLive supports remix, but remixing is an advanced feature that requires a good understanding of audio.

For example, in the output, you might want the one language to be in coding mode 5.1. You might want other languages to be available in coding mode 2.0.

Therefore you might choose to extract the following:

- Spanish in Dolby Digital 5.1
- French and English in AAC 2.0.

Assess captions

If you plan to include captions in an output group, you must determine if MediaLive can use the captions format in the source to produce the captions format that you want in the output.

Obtain the following information about the captions source.

Information to obtain	Verify the following
The available caption formats.	See the procedure after this table.
The available languages for each format.	

To assess the captions requirements

Follow these steps for each output group that you identified for your workflow.

1. Go to the section called "Captions: Supported formats" and find the section for the output group. For example, find the section called "HLS or MediaPackage output". In the table in that section, read down the first column to find the format (container) that the content provider is providing.
2. Read across to the *Source caption input* column to find the caption formats that MediaLive supports in that source format.
3. Then read across to the *Supported output captions* column to find the caption formats that MediaLive can convert the source format to.

You end up with a statement such as: "If you want to produce an HLS output and your source content is RTMP, you can convert embedded captions to burn-in, embedded, or WebVTT".

4. Verify that the source content from the content provider matches one of the formats in the *Supported caption input* column of the table. For example, verify that the source content contains embedded captions.
5. Find the list of captions formats that the downstream system supports. You obtained this list when you identified the encode requirements for the output groups that you identified. Verify that at least one of these output formats appears in the *Supported output captions* column of the table.

   If there is no match in the source content, or no match in the output, then you can't include captions in the output.

For example, assume that you need to produce an HLS output group. Assume that your content provider can give you content in RTP format with embedded captions. Assume that the downstream system requires that for HLS output, the output must include WebVTT captions.

Following the steps above, you read the table for HLS outputs. In the container column of the table, you find the row for RTP format. You read across to the source column and identify that embedded captions are a supported source format. You then read across to the output column and find that embedded captions can be converted to burn-in, embedded, or WebVTT captions. WebVTT captions is the format that the downstream system requires. Therefore, you conclude that you can include captions in the HLS output.

Step 5: Collect information about the source content

After you have assessed the source content and have identified suitable video, audio, and captions assets in that content, you must obtain information about those assets. The information you need is different for each type of source.

You don't need this information to create the input in MediaLive. But you will need this information when you attach the input to the channel in MediaLive.

Result of this step

After you have performed the procedures in this step, you should have source content information that looks like this example.

Example

Information	Format	Characteristics	Identifiers
Upstream System	RTP	with FEC	
Selected video	HEVC	1920x1080 5 Mbps maximum	PID 600
Selected audio	Dolby Digital 5.1		Spanish in PID 720
	AAC 2.0		Spanish in PID 746
	AAC 2.0		French in PID 747
	AAC 2.0		English in PID 759
Selected captions	Embedded		C1 = Spanish
			C2 = French
			C4 = English
	Teletext	10 languages	PID 815

Topics

- Identifying content in a CDI source
- Identifying content in an AWS Elemental Link source
- Identifying content in an HLS source
- Identifying content in a MediaConnect source
- Identifying content in an MP4 source
- Identifying content in an RTMP source
- Identifying content in an RTP source

Identifying content in a CDI source

The content in a CDI source always consists of uncompressed video, uncompressed audio, and captions.

Obtain identifying information from the content provider:

- For video – You don't need identifying information. MediaLive always extracts the first video that it encounters.
- For audio – The source might include multiple audio tracks, typically, one for each language. Obtain the track number of each audio asset to extract.
- For captions – The source might contain captions in its ancillary data. The supported caption types are ARIB, embedded (EIA-608 or CEA-708), and Teletext (OP47):
  - ARIB captions – You don't need any information. With ARIB captions, MediaLive extracts all the languages.
  - For embedded captions, obtain the languages in the channel numbers. For example, "channel 1 is French".
  - For Teletext captions, if your plan for captions is to convert the captions to a different format, you must obtain the page numbers for the languages that you want to convert. If you plan to pass through the captions as Teletext in the output, you don't need any identifiers.

Identifying content in an AWS Elemental Link source

The content in an AWS Elemental Link source is always a transport stream (TS) that contains one video asset and one audio pair. It might also contain embedded-style or object-style captions.

Obtain identifying information from the content provider:

- For video – You don't need identifying information.
- For audio – Obtain the languages in the tracks. For example, "track 1 is French".
- For captions – Obtain the identifiers:
  - For embedded captions, obtain the languages in the channel numbers. For example, "channel 1 is French".
  - For Teletext captions, if your plan for captions is to convert the captions to a different format, you must obtain the page numbers for the languages that you want to convert. If you plan to pass through the captions as Teletext in the output, you don't need any identifiers.

Also obtain the following information about the content:

- The maximum bitrate. You will have the option to throttle this bitrate when you set up the device in MediaLive. For more information, see *Setup: AWS Elemental Link* .
- Whether the content includes an embedded timecode. If it does, you can choose to use that timecode. For more information, see Timecode configurationthe section called “Timecodes”.
- Whether the content includes ad avail messages (SCTE-104 messages that MediaLive will automatically convert to SCTE-35 messages). For more information about ad avail messages, see the section called “SCTE-35 message processing”.

Identifying content in an HLS source

The content in an HLS container is always a transport stream (TS) that contains only one video rendition (program).

Obtain identifying information from the content provider:

- For video – Obtain the bitrates of the available video renditions.
- For audio – Obtain the PIDs or three-character language codes of the languages that you want.

> **Note**
>
> We recommend that you obtain the PIDs for the audio assets. They are a more reliable way of identifying an audio asset. Use the language only if there is only one instance of each audio language in the audio asset.

- For captions – Obtain the languages in the channel numbers. For example, "channel 1 is French". The captions are always embedded-style, containing up to four languages, each with its own channel number.

Identifying content in a MediaConnect source

The content in an AWS Elemental MediaConnect source is always a transport stream (TS). The TS is made up of

one program (SPTS) or multiple programs (MPTS). Each program contains a combination of video and audio. It might also contain embedded-style or object-style captions.

Obtain identifying information from the content provider:

- For video – Choose the one video program that you want, and obtain its PID or bitrate. (The source content might contain more than one video program.)

  If two video programs are identical, look at the audios and captions in each program. Those might be different, in which case you should choose the video program that contains the audio or captions formats you want.
- For audio – For the same program as the video, obtain the PIDs or three-character language codes of the audio languages that you want.

  > **Note**
  >
  > We recommend that you obtain the PIDs for the audio assets. They are a more reliable way of identifying an audio asset. Use the language only if there is only one instance of each audio language in the audio asset.
- For captions – For the same program as the video, obtain the identifiers:
  - If the captions are embedded, obtain the languages in the channel numbers. For example, "channel 1 is French".
  - If the captions are object-style captions (for example, DVB-Sub), obtain the PIDs of the captions languages that you want.

Identifying content in an MP4 source

The content in an MP4 source always consists of one video track, and one or more audio tracks. It might also contain embedded-style captions.

Obtain identifying information from the content provider:

- For audio – Obtain the track numbers or three-character language codes of the languages that you want.

> **Note**
>
> We recommend that you obtain the tracks for the audio assets. They are a more reliable way of identifying an audio asset. Use the language only if there is only one instance of each audio language in the audio asset.

- For captions – Obtain the languages in the channel numbers. For example, "channel 1 is French". The captions are always embedded-style, containing up to four languages, each with its own channel number. The captions might be embedded in the video track or might be embedded in an ancillary track.

Identifying content in an RTMP source

This procedure applies to both RTMP push and pull inputs from the internet, and to RTMP inputs from Amazon Virtual Private Cloud. The content in an RTMP input always consists of one video and one audio. It might also contain embedded-style captions.

Obtain identifying information from the content provider:

- For video – You don't need identifying information. MediaLive always extracts the single video asset.
- For audio – You don't need identifying information. MediaLive always extracts the single audio asset

- For captions – Obtain the languages in the channel numbers. For example, "channel 1 is French". The captions are always embedded-style, containing up to four languages, each with its own channel number. The captions might be embedded in the video track or might be embedded in an ancillary track.

Identifying content in an RTP source

This procedure applies to both RTP inputs from the internet and inputs from Amazon Virtual Private Cloud. The content in an RTP input is always a transport stream (TS). The TS is made up of one program (SPTS) or multiple programs (MPTS). Each program contains a combination of video and audio. It might also contain embedded-style or object-style captions.

Obtain identifying information from the content provider:

- For video – Choose the one video rendition that you want, and obtain its PID or bitrate. (The source content might contain more than one video rendition.)

  If two video renditions are identical, look at the audios and captions in each program. Those might be different, in which case you should choose the video rendition that contains the audio or captions formats you want.
- For audio – For the same rendition as the video, obtain the PIDs or three-character language codes of the audio languages that you want.

  > **Note**
  >
  > We recommend that you obtain the PIDs for the audio assets. They are a more reliable way of identifying an audio asset. Use the language only if there is only one instance of each audio language in the audio asset.
- For captions – For the same rendition as the video, obtain the identifiers:
  - If the captions are embedded, obtain the languages in the channel numbers. For example, "channel 1 is French".
  - If the captions are object-style captions (for example, DVB-Sub), obtain the PIDs of the captions languages that you want.

Step 6: Arrange for delivery to downstream systems

As the final step in preparing the downstream and upstream systems in your workflow, you must perform this step on each downstream system:

- You and the operator at the downstream system must agree on some portions of the path from AWS Elemental MediaLive to the downstream system.
- You must arrange for the operator at the downstream system to perform some setup so that MediaLive can successfully send outputs to these systems.

The setup is different for each type of output group and downstream system.

The *output* from MediaLive is considered *input* to this downstream system. You and the operator of the downstream system must agree about the input locations on the downstream system now, because when you create the MediaLive channel you need those URL locations.

Note that this guide describes how to set up an origin server. It does not describe how to set up the CDN that is downstream of the origin server. For information about that setup, see the documentation for the chosen origin server.

The options

The following table summarizes the combinations of output group and downstream system that are covered in the following sections.

Output group	Downstream system	Section to read
Archive	Amazon S3	the section called “Archive or frame capture”
CMAF Ingest	AWS Elemental MediaPackage	the section called “CMAF Ingest to MediaPackage”
Frame capture	Amazon S3	the section called “Archive or frame capture”
HLS	Amazon S3	the section called “HLS to Amazon S3”
HLS	AWS Elemental MediaStore	the section called “HLS to MediaStore”
HLS	AWS Elemental MediaPackage	the section called “HLS to MediaPackage”
HLS	AWS Elemental MediaPackage	the section called “HLS to MediaPackage v2”
HLS	HTTP server or Akamai CDN	the section called “HLS to HTTP”

Output group	Downstream system	Section to read
MediaPackage	AWS Elemental MediaPackage	the section called “MediaPackage ”
Microsoft Smooth	HTTP server	the section called “Microsoft Smooth”
RTMP	RTMP server	the section called “RTMP”
UDP	UDP address	the section called “UDP”

Result of this step

At the end of this step, you will have completed all the steps to prepare the downstream system to receive content from MediaLive.

Topics

- Archive or frame capture output group
- CMAF Ingest output group to MediaPackage
- HLS output group to Amazon S3
- HLS output group to MediaStore
- HLS output group to MediaPackage
- HLS output group to MediaPackage v2
- HLS output group to HTTP
- MediaPackage output group
- Microsoft Smooth output group
- RTMP output group
- UDP output group

Archive or frame capture output group

Follow this procedure if you determined that you will create an archive output group or frame capture output group. The destination for these output groups is always Amazon S3.

You and the operator of the downstream system must agree about the destination for the output of this output group. You will need this information when you create the MediaLive channel.

To arrange setup of the destination

1. Decide if you need two destinations for the output:
   - You need two destinations in a standard channel.
   - You need one destination in a single-pipeline channel.
2. Consult with the Amazon S3 user and decide on the bucket name or names. Ask the Amazon S3 user to create any buckets that don't already exist.
3. Discuss ownership with the Amazon S3 user. If the bucket belongs to another AWS account, you typically want that account to become the owner of the output. For more information, see the section called “Controlling access to the output”, after this procedure.
4. You can design the full destination paths now, or you can design them when you create the output group.

   If you want to design the full paths now, but you aren't familiar with the path requirements, see the section called “Destination fields” or the section called “Destination fields”.

   If you have two destinations, the destination paths must be different from each other in some way. At least one of the portions of one path must be different from the other. It is acceptable for all the portions to be different.
5. Make a note of the bucket or the full destination paths.

Note that you don't need user credentials to send to an S3 bucket. MediaLive has permission to write to the bucket via the trusted entity. Someone in your organization should have already set up these permissions. For more information, see the section called “Reference: summary of trusted entity access”.

Controlling access to the output

You might be sending output files to an Amazon S3 bucket that is owned by another AWS account. In this situation, you typically want the other account to become the owner of the output files (the object being put in the bucket). If the bucket owner doesn't become the object owner, you (MediaLive) will be the only agent that can delete the files when the files are no longer required.

It is therefore in everyone's interest to transfer ownership of the output files after they are in the Amazon S3 bucket.

To transfer object ownership, the following setup is required:

- The bucket owner must add a bucket permissions policy that grants you permission to add an Amazon S3 canned access control list (ACL) when MediaLive delivers the output files to the bucket. The bucket owner should read the information in Managing access with ACLs in the Amazon Simple Storage Service user guide. The bucket owner must set up ACL permissions for the bucket, not for the objects.
- The bucket owner should also set up object ownership. This feature effectively makes it mandatory (rather than optional) for the sender (MediaLive) to include the *Bucket owner full control* ACL. The bucket owner should read the information in Controlling object ownership in the Amazon Simple Storage Service user guide.

  If the bucket owner implements this feature, then you must set up MediaLive to include the ACL. If you don't, delivery to the Amazon S3 bucket will fail.
- You must set up MediaLive to include the *Bucket owner full control* ACL when it delivers to the bucket. You will perform this setup when you create the channel.

The S3 canned ACL feature supports ACLs other than *Bucket owner full control*. But those other ACLs are typically not applicable to the use case of delivering video from MediaLive.

CMAF Ingest output group to MediaPackage

Follow this procedure if you determined that you will create a CMAF Ingest output group in order to send CMAF content to AWS Elemental MediaPackage.

1. Decide if you need two destination URLs for the output:
   - You need two destinations in a standard channel.
   - You need one destination in a single-pipeline channel.
2. Obtain the one or two URLs. The MediaPackage terminology for the URL is *input endpoint*. Make sure that you obtain the URLs (which start with `https://`), not the channel name (which starts with `arn`).

   Note that you don't use user credentials to send to CMAF Ingest to MediaPackage.

Example

Two URLs look like this example:

```
https://mz82o4-1.ingest.hnycui.mediapackagev2.us-west-2.amazonaws.com/in/v1/curling-channel-group/1/curling-channel/
```

```
https://mz82o4-2.ingest.hnycui.mediapackagev2.us-west-2.amazonaws.com/in/v1/curling-channel-group/1/curling-channel/
```

Note the following:

- The `v1/` is the version of the MediaPackage destination URL schema, it doesn't refer to MediaPackage v1.
- `curling-channel-group/` is the name of the channel group that the MediaPackage operator created.
- `curling-channel/` is the name of the MediaPackage channel that the MediaPackage operator created. It isn't the name of the MediaLive channel.
- The only difference in the two URLs is the `-1` and `-2` before `.ingest`. and the `1/` and `2/` after the channel group.

HLS output group to Amazon S3

Follow this procedure if you determined that you will create an HLS output group with Amazon S3 as the destination.

You and the operator of the downstream system must agree about the destination for the output of the HLS output group. You will need this information when you create the MediaLive channel.

To arrange setup of the destination

1. Decide if you need two destinations for the output:
   - You need two destinations in a standard channel.
   - You need one destination in a single-pipeline channel.
2. Consult with the Amazon S3 user and decide on the bucket name or names. Ask the Amazon S3 user to create any buckets that don't already exist.

3. Discuss ownership with the Amazon S3 user. If the bucket belongs to another AWS account, you typically want that account to become the owner of the output. For more information, see the section called "Controlling access to the output", after this procedure.
4. You can design the full destination paths now, or you can design them when you create the output group.

   If you want to design the full paths now, but you aren't familiar with the path requirements, see the section called "Step 1: Design the path".

   If you have two destinations, the destination paths must be different from each other in some way. At least one of the portions of one path must be different from the other. It is acceptable for all the portions to be different.
5. Make a note of the bucket or full destination paths.

Note that you don't need user credentials to send to an S3 bucket. MediaLive has permission to write to the S3 bucket via the trusted entity. Someone in your organization should have already set up these permissions. For more information, see the section called "Reference: summary of trusted entity access".

Controlling access to the output

You might be sending output files to an Amazon S3 bucket that is owned by another AWS account. In this situation, you typically want the other account to become the owner of the output files (the object being put in the bucket). If the bucket owner doesn't become the object owner, you (MediaLive) will be the only agent that can delete the files when the files are no longer required.

It is therefore in everyone's interest to transfer ownership of the output files after they are in the Amazon S3 bucket.

To transfer object ownership, the following setup is required:

- The bucket owner must add a bucket permissions policy that grants you permission to add an Amazon S3 canned access control list (ACL) when MediaLive delivers the output files to the bucket. The bucket owner should read the information in Managing access with ACLs in the Amazon Simple Storage Service user guide. The bucket owner must set up ACL permissions for the bucket, not for the objects.
- The bucket owner should also set up object ownership. This feature effectively makes it mandatory (rather than optional) for the sender (MediaLive) to include the *Bucket owner full*

control ACL. The bucket owner should read the information in Controlling object ownership in the Amazon Simple Storage Service user guide.

If the bucket owner implements this feature, then you must set up MediaLive to include the ACL. If you don't, delivery to the Amazon S3 bucket will fail.

- You must set up MediaLive to include the *Bucket owner full control* ACL when it delivers to the bucket. You will perform this setup when you create the channel.

The S3 canned ACL feature supports ACLs other than *Bucket owner full control*, but those other ACLs are typicallly not applicable to the use case of delivering video from MediaLive.

HLS output group to MediaStore

Follow this procedure if you determined that you will create an HLS output group, with AWS Elemental MediaStore as the destination.

You and the operator of the downstream system must agree about the destination for the output of the HLS output group. You will need this information when you create the MediaLive channel.

To arrange setup of the destination

1. Decide if you need two destinations for the output:
   - You need two destinations in a standard channel.
   - You need one destination in a single-pipeline channel.
2. If you have two destinations, the destination paths must be different from each other in some way. At least one of the portions of one path must be different from the other. It is acceptable for all the portions to be different.

   You can design the full destination paths now, or you can decide only on the container name or names:

   - If you want to design the full paths now, but you aren't familiar with the destination requirements for an HLS output, see the section called “Step 1: Design the path”. You and the MediaStore user must agree on the containers that you want to use.
   - If you want to decide only on the containers, you and the MediaStore user must agree on which containers to use.
3. Ask the MediaStore user to create any containers that don't already exist.

4. Obtain the data endpoint for the container or containers. For example:

   ```
   https://a23f.data.mediastore.us-west-2.amazonaws.com
   ```

   ```
   https://fe30.data.mediastore.us-west-2.amazonaws.com
   ```

   You need the data endpoints. You don't need the container name.

Note that you don't need user credentials to send to MediaStore containers. MediaLive has permission to write to the MediaStore container via the trusted entity. Someone in your organization should have already set up these permissions. For more information, see the section called “Reference: summary of trusted entity access”.

HLS output group to MediaPackage

Follow this procedure if you determined that you will create an HLS output group, and will send to AWS Elemental MediaPackage over HTTPS. You can also send to AWS Elemental MediaPackage by creating a MediaPackage output group. For more information, seee the section called “HLS versus MediaPackage”.

You and the operator of the downstream system must agree about the destination for the output of the HLS output group. You will need this information when you create the MediaLive channel.

To arrange setup of the destination

1. Ask the MediaPackage user to create one channel on MediaPackage. Even if the MediaLive channel is a standard channel (with two pipelines), you need only one MediaPackage channel.
2. Arrange with the MediaPackage user to set up HTTPS user credentials. You must send to MediaPackage over a secure connection.
3. Obtain the following information:

   - The two URLs (input endpoints is the MediaPackage terminology) for the channel. The two URLs for a channel look like this:

     ```
     https://6d2c.mediapackage.uswest-2.amazonaws.com/in/v2/9dj8/9dj8/
     channel
     ```

     ```
     https://6d2c.mediapackage.uswest-2.amazonaws.com/in/v2/9dj8/e333/
     channel
     ```

The two URLs are always identical, except for the folder just before `channel`.

Make sure that you obtain the URLs (which start with `https://`), not the channel name (which starts with `arn`).

- The user name and password to access the downstream system, if the downstream system requires authenticated requests. Note that these user credentials relate to user authentication, not to the protocol. User authentication is about whether the downstream system will accept your request. The protocol is about whether the request is sent over a secure connection.

HLS output group to MediaPackage v2

Follow this procedure if you determined that you will create an HLS output group, and will send to MediaPackage v2. One use case for using version 2 is to implement a glass-to-glass low latency workflow that includes both MediaLive and MediaPackage.

You and the operator of the downstream system must agree about the destination for the output of the HLS output group. You will need this information when you create the MediaLive channel.

To arrange setup of the destination

1. Ask the MediaPackage user to create one channel on MediaPackage. Even if the MediaLive channel is a standard channel (with two pipelines), you need only one MediaPackage channel.
2. Obtain the two URLs (input endpoints is the MediaPackage terminology) for the channel. The two URLs for a channel look like this:

   ```
   https://mz82o4-1.ingest.hnycui.mediapackagev2.us-west-2.amazonaws.com/in/v1/live-sports/1/curling/index
   ```

   ```
   https://mz82o4-2.ingest.hnycui.mediapackagev2.us-west-2.amazonaws.com/in/v1/live-sports/2/curling/index
   ```

   The two URLs are slightly different, as shown in the examples above.

   Make sure that you obtain the URLs (which start with `https://`), not the channel name (which starts with `arn`).

   Note that you don't use user credentials in order to send to MediaPackage v2.

HLS output group to HTTP

Follow this procedure if you determined that you will create an HLS output group with one of the following downstream systems as the destination:

- An HTTP or HTTPS PUT server.
- An HTTP or HTTPS WebDAV server.
- An Akamai origin server.

You and the operator of the downstream system must agree about the destination for the output of the HLS output group. You will need this information when you create the MediaLive channel.

When you deliver HLS over HTTP, you are often delivering to an origin server. The origin server typically has clear guidelines about the rules for the destination path, including the file name of the main manifest (the `.M3U8` file).

To arrange setup of the destination

You must talk to the operator at the downstream system to coordinate your setup.

1. If the downstream system isn't an Akamai server, find out if it uses PUT or WebDAV.
2. Find out if the downstream system has special connection requirements. These connection fields are grouped in the console in the **CDN settings** section for the HLS output group. To display this page on the MediaLive console, in the **Create channel** page, in the **Output groups** section, choose **Add**, then choose **HLS**. Choose the group, then in **HLS settings**, open **CDN settings**.
3. Decide if you need two destinations for the output:
   - You need two destinations in a standard channel.
   - You need one destination in a single-pipeline channel.
4. Find out if the downstream system uses a secure connection. If it does, arrange with the operator to set up user credentials.
5. Find out if the downstream system requires custom paths inside the main manifests and the child manifests. For more information, see the section called “Manifests – custom HLS manifest paths”.
6. If you are setting up a standard channel, find out if the downstream system supports redundant manifests. If so, decide if you want to implement this feature. For more

information, see the section called “Manifests – Redundant HLS manifests”, and specifically the section called “Rules for most systems” and the section called “Rules for Akamai” for specific instructions.

7. Talk to the operator at the downstream system to agree on a full destination path for the three categories of HLS files (the main manifests, the child manifests, and the media files). MediaLive always puts all three categories of files for each destination in this one location. It’s not possible to configure MediaLive to put some files in another location.

   If you have two destinations, the destination paths must be different from each other in some way. At least one of the portions of one path must be different from the other. It is acceptable for all the portions to be different. Discuss this requirement with the operator of the downstream system. The downstream system might have specific rules about uniqueness.

8. Talk to the operator at the downstream system about special requirements for the names of the three categories of HLS files. Typically, the downstream system doesn’t have special requirements.

9. Talk to the operator at the downstream system about special requirements for the modifier on the names of the child manifests and media files.

   The child manifests and media files always include this modifier in their file names. This modifier distinguishes each output from the other, so it must be unique in each output. For example, the files for the high-resolution output must have a different name from the files for the low-resolution output. For example, the files for one output could have the file name and modifier `curling_high`, while the other output could have `curling_low`.

   Typically, the downstream system doesn’t have special requirements.

10. Ask the operator of the downstream system if the media files should be set up in separate subdirectories. For example, one subdirectory for the first 1000 segments, another subdirectory for the second 1000 segments, and so on.

    Most downstream systems don’t require separate subdirectories.

11. Agree on the portions of the destination path where the downstream system has special requirements.

    - For example, the downstream system might only require that you send to a specific host. The downstream system doesn't need to know about the folder or file names you will use.

For example, send to two folders that you name, but on the host at `https://203.0.113.55`

Or send to two folders that you name, but on the hosts at `https://203.0.113.55` and `https://203.0.113.82`

- Or the downstream system might require a specific host and folder, but with a file name that you choose. For example, this host and folders:

  `https://203.0.113.55/sports/delivery/`

  `https://203.0.113.55/sports/backup/`

12. Make a note of the information you have collected:

- The connection type for the downstream system – Akamai, PUT, or WebDAV.
- The settings for connection fields, if the downstream system has special requirements.
- The protocol for delivery—HTTP or HTTPS.
- The user name and password to access the downstream system, if the downstream system requires authenticated requests. Note that these user credentials relate to user authentication, not to the protocol. User authentication is about whether the downstream system will accept your request. The protocol is about whether the request is sent over a secure connection.
- All or part of the destination paths, possibly including the file names.
- Whether you need to set up separate subdirectories.

MediaPackage output group

Follow this procedure if you determined that you will create a MediaPackage output group.

You and the operator of the downstream system must agree about the destination for the output of the MediaPackage output group. You will need this information when you create the MediaLive channel.

Note that you can send to AWS Elemental MediaPackage by creating a MediaPackage output group, or by creating an HLS output group. See the section called “HLS versus MediaPackage” for a description of the differences. This section describes the first option.

To arrange setup of the destination

1. Ask the MediaPackage user to create one channel. Even if the MediaLive channel is a standard channel (with two pipelines), you need only one MediaPackage channel.
2. Obtain the ID of the MediaPackage channel. The channel ID is case sensitive.

Note that you don't need user credentials to send a MediaPackage output to MediaPackage. MediaLive has permission to write to MediaPackage via the trusted entity. Someone in your organization should have already set up these permissions. For more information, see the section called “Reference: summary of trusted entity access”.

Microsoft Smooth output group

Follow this procedure if you have determined that you will create a Microsoft Smooth output group.

You and the operator of the downstream system must agree about the destination for the output of the Microsoft Smooth output group. You will need this information when you create the MediaLive channel.

To arrange setup of the destination

1. Decide if you need two destinations for the output:
   - You need two destinations in a standard channel.
   - You need one destination in a single-pipeline channel.
2. Talk to the operator at the Microsoft IIS server to agree on a full path for the output. Make a note of the URLs that you agree on. For example:

   ```
   https://203.0.113.55/sports/curling

   https://203.0.113.82/sports/curling
   ```
3. Arrange with the operator to set up user credentials, if the protocol is HTTPS.
4. Find out if the downstream system has special connection requirements. These connection fields are in the **General configuration** section for the Microsoft Smooth output group. To display this page on the MediaLive console, in the **Create channel** page, in **Output groups** section, choose **Add**, then choose **Microsoft Smooth**. Choose the group, then in **Microsoft Smooth settings**, open **General configuration**.

5. Make a note of the information you have collected:

   - The URLs.
   - The user name and password to access the Microsoft IIS servers, if the server requires authenticated requests. Note that these user credentials relate to user authentication, not to the protocol. User authentication is about whether the server will accept your request. The protocol is about whether the request is sent over a secure connection.
   - The settings for connection fields, if the downstream system has special requirements.

RTMP output group

Follow this procedure if you determined that you will create an RTMP output group.

You and the operator of the downstream system must agree about the destination for the output of the RTMP output group. You will need this information when you create the MediaLive channel.

To arrange setup of the destination

1. If the RTMP server is a social media site, the host of the site might have instructions that can supplement the following information. Obtain these instructions.
2. Decide if you need two destinations for the output:

   - If the MediaLive channel is a standard channel, you need two destinations.
   - If the MediaLive channel is a single-pipeline channel, you need one destination.
3. Make sure that the RTMP operator sets up to expect MediaLive output at one or two inputs on the RTMP server, as appropriate.
4. Obtain the following information from the RTMP operator:

   - The protocol for MediaLive to use—RTMP or RTMPS.
   - The user name and password to access the downstream system, if the downstream system requires authenticated requests. Note that these user credentials relate to user authentication, not to the protocol. User authentication is about whether the downstream system will accept your request. The protocol is about whether the request is sent over a secure connection.
   - IP address.
   - Port number.

- Application name. Also called *app name*.
- Stream name. Also called *application instance* or *app instance* or *stream key*.

  The operator might give you the application name and stream name as separate pieces of data. Or they might give you a complete path in the format **`string/string`**. In this case, the first string is the application name and the second string is the stream name.

Here is an example of the information that the operator will give you:

`rtmp://203.0.113.28:80/xyz/ywq7b`

`rtmp://203.0.113.17:80/xyz/ywq7b`

Where `xyz` is the application name, and `ywq7b` is the stream name.

In this example, the two URLs have different IP addresses but the same application name/stream name portion. Your RTMP server might follow a different rule.

UDP output group

Follow this procedure if you have determined that you will create a UDP output group.

You and the operator of the downstream system must agree about the destination for the output of the UDP output group. You will need this information when you create the MediaLive channel.

To arrange setup of the destination

1. Decide if you need two destinations for the output:
   - If the MediaLive channel is a standard channel, you need two destinations.
   - If the MediaLive channel is a single-pipeline channel, you need one destination.
2. Speak to the operator who manages the downstream system that will receive UDP content. Make sure that the operator sets up to expect one or two MediaLive outputs, as appropriate.
3. Obtain the following information from the operator:
   - Whether the protocol is UDP or RTP
   - The URLs
   - The port numbers

Each URL will look like this, for example:

udp://203.0.113.28:5000

udp://203.0.113.33:5005

Note that in this example, the port numbers are not sequential. These non-sequential numbers are important if you plan to enable FEC in the outputs (this field is in the **Output** pane of the UDP output group). With FEC, you must leave space between the port numbers for the two destinations. For example, if one destination is **rtp://203.0.113.28:5000**, assume that FEC also uses port 5002 and 5004. So the lowest possible port number for the other destination is 5005.

Next steps

You have now planned your workflow:

- You started by identifying the outputs that you require.
- You then worked back to the source. You assessed the upstream system, and obtained information about the sources.

- You identified the type of MediaLive input to create.
- You then returned to the output side and coordinated with the downstream system for delivery of the outputs that MediaLive will produce.

You now have the information that you need to perform these steps:

- Design the channel that is the key resource in your workflow. See *Setup: Planning the channel*.
- Create the appropriate MediaLive inputs, as described in *Setup: Creating inputs*.

Planning the channel in the MediaLive workflow

You should plan the AWS Elemental MediaLive channel as the second stage of planning a transcoding *workflow*. You should have already performed the first stage of setting up the upstream and downstream systems, as described in *Setup: Preparing upstream and downstream*.

The channel provides the ability to configure for different characteristics of the outputs, and for including a wide array of video features. But before you plan these details, you should plan the basic features for the channel.

This chapter describes how to plan these basic features. If you take the time to carefully plan these features of the channel, the job of creating the channel will proceed more smoothly.

> **Note**
>
> On the output side, we refer to each video or audio or caption stream, track, or program as an *encode*.

Topics

Step 1: Identify the output encodes

When you prepared the downstream systems, you identified the output groups that you need. Now, as part of the planning of the channel, you must identify the encodes to include in each output group you have decided to create. An *encode* refers to the audio, video, or captions streams in the output.

Topics

- Identify the captions encodes
- Summary of encode rules for output groups
- Example of a plan for output encodes

Identify the video encodes

You must decide on the number of video encodes and their codecs. Follow this procedure for each output group.

1. Determine the maximum number of encodes that are allowed in the output group. The following rules apply for each type of output group.

Type of output group	Rule for video encodes
Archive	One video encode.
CMAF Ingest	One or more video encodes. Typically, there are multiple video encodes.
Frame Capture	One video encode.
HLS or MediaPackage	One or more video encodes. Typically, there are multiple video encodes.
Microsoft Smooth	One or more video encodes. Typically, there are multiple video encodes.
RTMP	One video encode.
UDP	One video encode.

2. If the output group allows more than one video encode, decide how many you want. Keep in mind that you can create multiple output encodes from the single video source that MediaLive ingests.
3. Identify the codec or codecs for the video encodes.
   - For most types of output groups, the downstream system dictates the codec for each video encode, so you obtained this information when you identified the output encodes.

- For an archive output group, you decide which codec suits your purposes.

4. Identify the resolution and bitrate for each video encode. You might have obtained requirements or recommendations from your downstream system when you identified the output encodes.
5. Identify the frame rates for each video encode. If you are using more than one video encode, you can ensure compatibility by choosing output frame rates that are multiples of the lowest frame rate used.

   Examples:

   - 29.97 and 59.94 frames per second are compatible frame rates.
   - 15, 30, and 60 frames per second are compatible frame rates.
   - 29.97 and 30 frames per second are *not* compatible frame rates.
   - 30 and 59.94 frames per second are *not* compatible frame rates.

Identify the audio encodes

You must decide on the number of audio encodes. Follow this procedure for each output group.

1. Determine the maximum number of encodes that are allowed in the output group. The following rules apply for each type of output group.

Type of output group	Rule for audio encodes
Archive	Zero or more audio encodes.
CMAF Ingest	Zero or more audio encodes. Typically, there are multiple audio encodes.
Frame Capture	Zero audio encodes.
HLS or MediaPackage	Zero or more audio encodes. Typically, there are multiple audio encodes.
Microsoft Smooth	Zero or more audio encodes. Typically, there are multiple audio encodes.

Type of output group	Rule for audio encodes
RTMP	Zero or one audio encodes.
UDP	One or more audio encodes.

2. If the output group allows more than one audio encode, decide how many you want. These guidelines apply:

   - Each different combination of output codec, coding mode, and language is one encode.

     MediaLive can produce a specific coding mode only if the source contains that coding mode or a higher mode. For example, MediaLive can create 1.0 from a 1.0 or a 2.0 source. It can't create 5.1 from a 2.0 source.
   - MediaLive can produce a specific language only if the source contains that language.
   - MediaLive can produce more than one encode for a given language.

     For example, you could choose to include Spanish in Dolby 5.1 and in AAC 2.0.

   - There is no requirement for the count of encodes to be the same for all languages. For example, you could create two encodes for Spanish, and only one encode for the other languages.
3. Identify the bitrate for each audio encode. You might have obtained requirements or recommendations from your downstream system when you identified the output encodes.

Identify the captions encodes

You must decide on the number of captions encodes. Follow this procedure for each output group.

1. Determine the maximum number of captions encodes that are allowed in the output group. The following rules apply for each type of output group.

Type of output group	Rule for captions encodes
Archive	Zero or more captions encodes. The captions are either embedded or object-style captions.

Type of output group	Rule for captions encodes
CMAF Ingest	Zero or more captions encodes. Typically, there are caption languages to match the audio languages. The captions are always sidecar captions.
Frame Capture	Zero captions encodes.
HLS or MediaPackage	Zero or more captions encodes. Typically, there are caption languages to match the audio languages. The captions are either embedded or sidecar captions.
Microsoft Smooth	Zero or more captions encodes. Typically, there are caption languages to match the audio languages. The captions are always sidecar captions.
RTMP	Zero or one caption encodes. The captions are either embedded or object-style captions.
UDP	One or more captions encodes. The captions are either embedded or object-style captions.

2. Identify the category that each caption format belongs to. See the list in the section called "Captions categories". For example, WebVTT captions are sidecar captions.
3. Use this category to identify the number of captions encodes you need in the output group.
   - For embedded captions, you always create one captions encode.
   - For object-style captions and sidecar captions, you create one captions encode for each format and language that you want to include.

Summary of encode rules for output groups

This table summarizes the rules for encodes for each output group. In the first column, find the output you want, then read across the row.

Type of output group	Rule for video encodes	Rule for audio encodes	Rule for captions encodes
Archive	One video encode.	Zero or more audio encodes.	Zero or more captions encodes. The captions are either embedded or object-style captions.
CMAF Ingest	One or more video encodes. Typically , there are multiple video encodes.	Zero or more audio encodes. Typically , there are multiple audio encodes.	Zero or more captions encodes. Typically, there are caption languages to match the audio languages. The captions are sidecar captions.
Frame Capture	One video encode.	Zero audio encodes.	Zero captions encodes.
HLS or MediaPackage	One or more video encodes. Typically , there are multiple video encodes.	Zero or more audio encodes. Typically , there are multiple audio encodes.	Zero or more captions encodes. Typically, there are caption languages to match the audio languages. The captions are either embedded or sidecar captions.
Microsoft Smooth	One or more video encodes. Typically	Zero or more audio encodes. Typically	Zero or more captions encodes.

Type of output group	Rule for video encodes	Rule for audio encodes	Rule for captions encodes
	, there are multiple video encodes.	, there are multiple audio encodes.	Typically, there are caption languages to match the audio languages. The captions are always sidecar captions.
RTMP	One video encode.	Zero or one audio encodes.	Zero or one caption encodes. The captions are either embedded or object-style captions.
UDP	One video encode.	One or more audio encodes.	One or more captions encodes. The captions are either embedded or object-style captions.

Some output groups also support audio-only outputs. See the section called “Output groups and outputs”.

Some output groups also support outputs that contain JPEG files, to support trick play according to the Roku specification. See the section called “Trick-play track via the Image Media Playlist specification”.

Example of a plan for output encodes

After you have performed this procedure, you should have information that looks like this example.

Example

Output group	Type of encode	Encode nickname	Characteristics of the encode
HLS	Video	VideoA	AVC 1920x1080, 5 Mbps
		VideoB	AVC 1280x720, 3 Mbps
		VideoC	AVC 320x240, 750 Kbps
	Audio	AudioA	AAC 2.0 in English at 192000 bps
		AudioB	AAC 2.0 in French at 192000 bps
	Captions	CaptionsA	WebVTT (object-style) converted from embedded, in English
		CaptionsB	WebVTT (object-style) converted from embedded, in French
RTMP	Video	VideoD	AVC 1920x1080, 5Mbps
	Audio	AudioC	Dolby Digital 5.1 in Spanish
	Captions	CaptionsC	RTMP CaptionInfo (converted from embedded) in Spanish

Output group	Type of encode	Encode nickname	Characteristics of the encode
Archive	Video	VideoE	AVC, 1920x1080, 8.5 Mbps
	Audio	AudioD	Dolby Digital 2.0 in Spanish
		AudioE	Dolby Digital 2.0 in French
		AudioF	Dolby Digital 2.0 in English
	Captions	CaptionsD	DVB-Sub (object-style) converted from Teletext, in 6 languages

Step 2: Map the output encodes to the sources

In the first step of planning the channel, you identified the number of encodes you need in each output group. You must now determine which assets from the source you can use to produce those encodes.

Result of this procedure

After you have performed this procedure, you will have identified the following key components that you will create in the channel:

- The video input selectors
- The audio input selectors
- The captions input selectors

Identifying these components is the last step in planning the *input* side of the channel.

To map the output to the sources

1. Obtain the *list of output encodes* you want to produce. You created this list in the previous step. It is useful to organize this list into a table. For example:

   Example

Output group	Type of encode	Encode nickname	Characteristics of the encode
HLS	Video	VideoA	AVC 1920x1080, 5 Mbps
		VideoB	AVC 1280x720, 3 Mbps
		VideoC	AVC 320x240, 750 Kbps
	Audio	AudioA	AAC 2.0 in English at 192000 bps
		AudioB	AAC 2.0 in French at 192000 bps
	Captions	CaptionsA	WebVTT (object-style) converted from embedded, in English
		CaptionsB	WebVTT (object-style) converted from embedded, in French
RTMP	Video	VideoD	AVC 1920x1080, 5Mbps
	Audio	AudioC	Dolby Digital 5.1 in Spanish

Output group	Type of encode	Encode nickname	Characteristics of the encode
	Captions	CaptionsC	RTMP CaptionInfo (converted from embedded) in Spanish
Archive	Video	VideoE	AVC, 1920x1080, 8.5 Mbps
	Audio	AudioD	Dolby Digital 2.0 in Spanish
		AudioE	Dolby Digital 2.0 in French
		AudioF	Dolby Digital 2.0 in English
	Captions	CaptionsD	DVB-Sub (object-style) converted from Teletext, in 6 languages.

2. Obtain the *list of sources* that you created when you assessed the source content and collected identifiers. For an example of such a list, see the section called “Step 4: Assess the upstream system ”.
3. In your table of output encodes, add two more columns, labeled *Source* and *Identifier in source*.
4. For each encode (column 2), find a line in the *list of sources* that can produce that encode. Add the source codec and the identifier of that source codec. This example shows a completed table.

Example

Output group	Type of encode	Encode nickname	Characteristics of the encode	Source	Identifier in source
HLS	Video	VideoA	AVC 1920x1080, 5 Mbps	HEVC	PID 600
		VideoB	AVC 1280x720, 3 Mbps	HEVC	PID 600
		VideoC	AVC 320x240, 750 Kbps	HEVC	PID 600
	Audio	AudioA	AAC 2.0 in English at 192000 bps	AAC 2.0	PID 759
		AudioB	AAC 2.0 in French at 192000 bps	AAC 2.0	PID 747
	Captions	CaptionsA	WebVTT (object-style) converted from embedded, in English	Embedded	Channel 4

Output group	Type of encode	Encode nickname	Characteristics of the encode	Source	Identifier in source
		CaptionsB	WebVTT (object-style) converted from embedded, in French	Embedded	Channel 2
RTMP	Video	VideoD	AVC 1920x1080, 5Mbps	HEVC	PID 600
	Audio	AudioC	Dolby Digital 5.1 in Spanish	Dolby Digital 5.1	PID 720
	Captions	CaptionsC	RTMP CaptionInfo (converted from embedded) in Spanish	Embedded	Channel 3
Archive	Video	VideoE	AVC, 1920x1080, 5 Mbps	HEVC	PID 600
	Audio	AudioD	Dolby Digital 2.0 in Spanish	AAC 2.0	PID 746
		AudioE	Dolby Digital 2.0 in French	AAC 2.0	PID 747

Output group	Type of encode	Encode nickname	Characteristics of the encode	Source	Identifier in source
		AudioF	Dolby Digital 2.0 in English	AAC 2.0	PID 759
	Captions	CaptionsD	DVB-Sub (object-style) converted from Teletext, in 6 languages.	Teletext	PID 815

You will use this information when you create the channel:

- You will use the source and source identifier information when you create the input selectors.
- You will use the characteristics information when you create the encodes in the output groups.

5. After you have identified the source assets, group those assets that are being used more than once, to remove the duplicates.
6. Label each asset by its type—video, audio, or captions.

Example

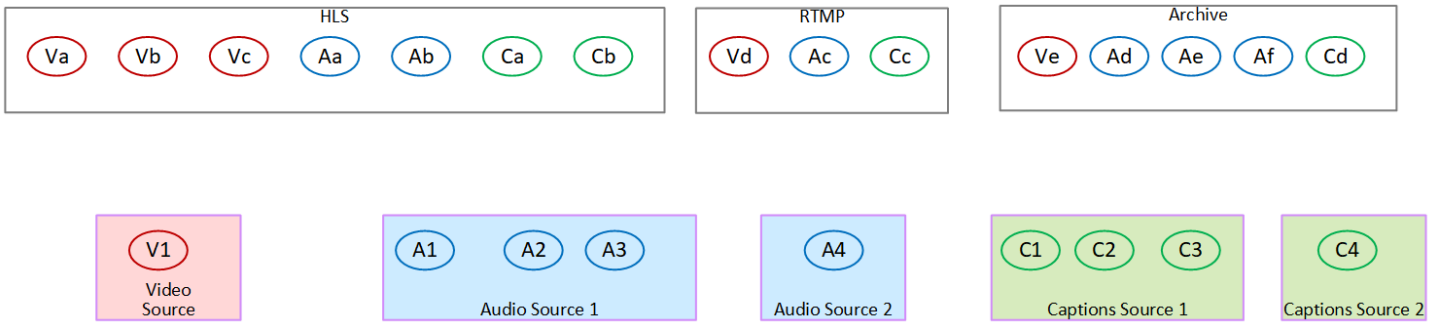
Input asset	Asset nickname	Source	Characteristics	Identifier in Source
video 1	Video1	Video	HEVC	PID 600
audio 1	Audio1	Audio	AAC 2.0 Spanish	PID 746
audio 2	Audio2		AAC 2.0 French	PID 747

Input asset	Asset nickname	Source	Characteristics	Identifier in Source
audio 3	Audio3		AAC 2.0 English	PID 759
audio 4	Audio4		Dolby Digital 5.1 Spanish	PID 720
captions 1	Captions1	Captions	Embedded French	Channel 2
captions 2	Captions2		Embedded Spanish	Channel 3
captions 3	Captions3		Embedded English	Channel 4
captions 4	Captions4		Teletext, all languages	PID 815

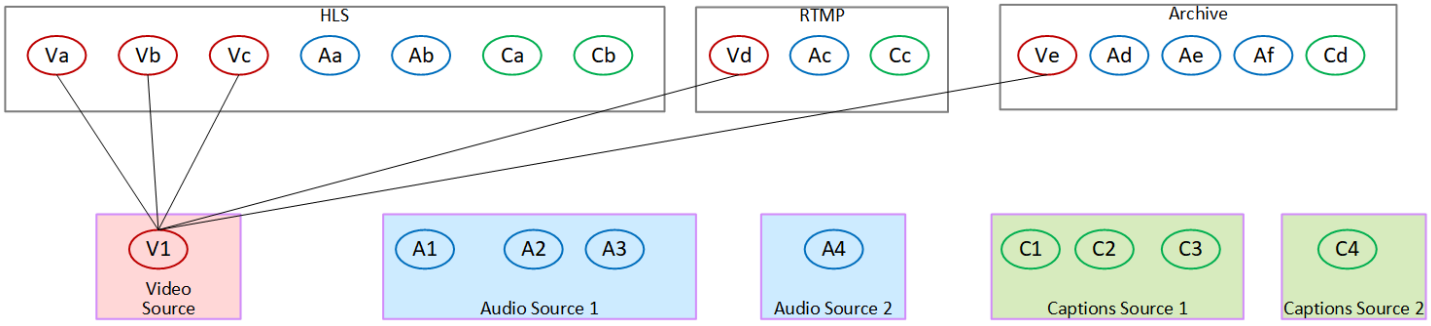
Example of mapping

The following diagrams illustrate the mapping of the output encodes back to source assets. The first diagram shows the outputs (at the top) and the sources (at the bottom). The other three diagrams shows the same outputs and sources with the mappings for video, for audio, and for captions.

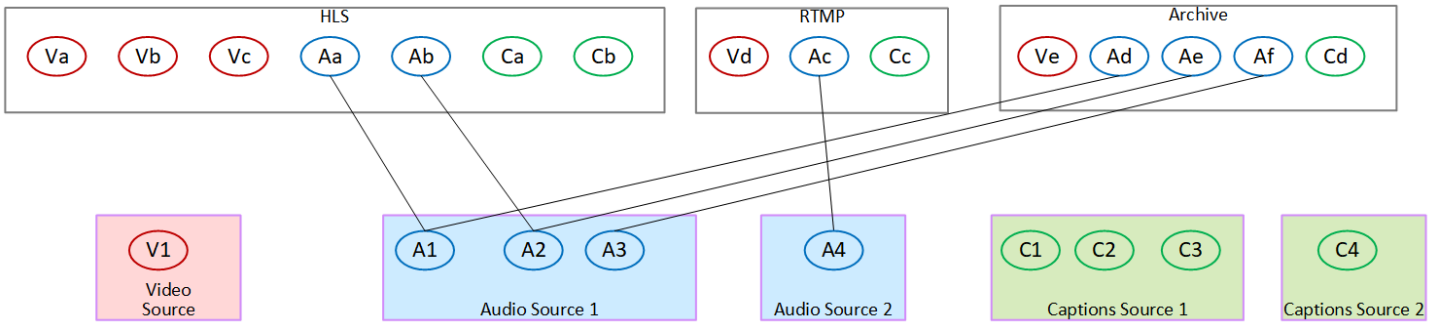
Encodes and assets



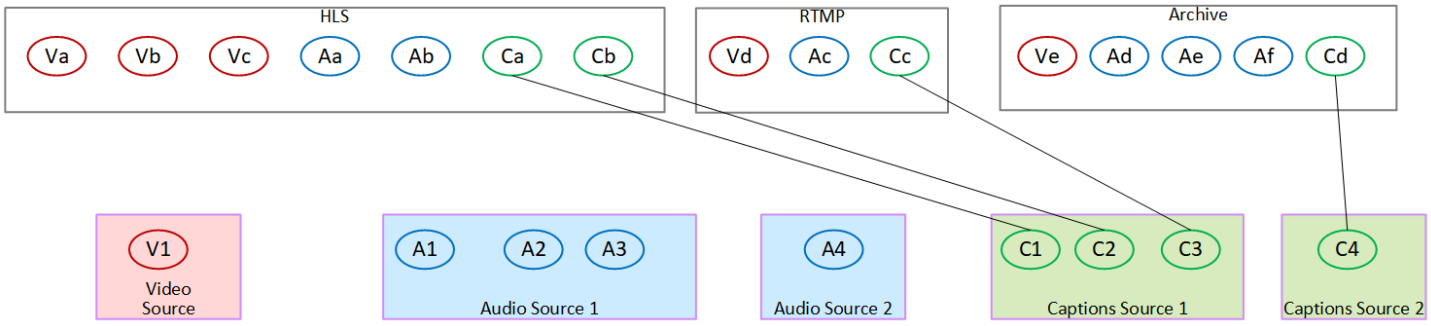
Mapping video encodes to assets



Mapping audio encodes to assets



Mapping captions encodes to assets



Step 3: Design the output groups

In the first step of planning the channel, you identified the video, audio, and captions encodes to include in each output group.

You must now *organize* these video, audio, and captions encodes into outputs in each output group. You must organize these encodes to follow the rules that each type of output group dictates.

Result of this procedure

After you have performed this procedure, you will have designs for the following:

- The organization of the outputs in each output group.
- The organization of the video, audio, and captions encodes in each output.

You have now planned the *output* side of the channel.

AWS Elemental MediaLive compared to AWS Elemental Live

If you are familiar with AWS Elemental Live, note that AWS Elemental Live refers to output *streams*, while MediaLive refers to *encodes*. Apart from that, the concepts are the same: MediaLive channels combine video, audio, and captions encodes into outputs, and outputs are placed in output groups.

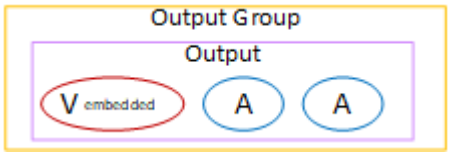
Topics

Organize encodes in an Archive output group

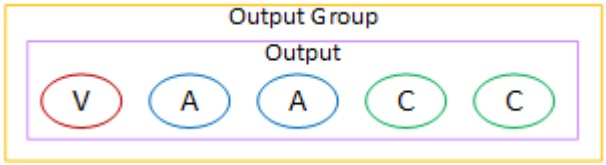
An Archive output group can contain one video encode, one or more audio encodes, and one or more captions encodes (either embedded or object-style).

Plan for the output group to contain one output that contains all the encodes.

This diagram illustrates an Archive output group that contains one output that holds one video encode with embedded captions, and two audio encodes.



This diagram illustrates an Archive output group that contains one output that holds one video encode, two audio encodes, and two object-style captions encode.



Organize encodes in a CMAF Ingest output group

A CMAF Ingest output group contains one video encode, one audio encode, and one captions encode.

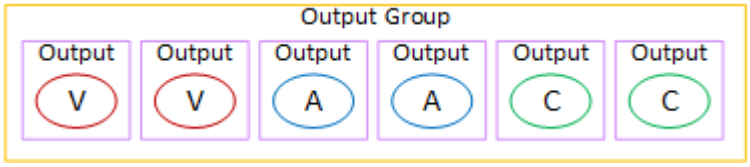
Plan for the output group to contain the following:

- One output for each video encode.
- One output for each audio encode.

  The audio encodes might be for different languages, or they might be for different bitrates, or they might be for different languages and bitrates.
- One output for each captions encode. The captions in a CMAF Ingest output group are always sidecar captions.

The arrangement of the audio encodes in this output group is called an *audio rendition group*.

This diagram illustrates a CMAF Ingest output group with an audio rendition group.



Organize encodes in a Frame Capture output group

A frame capture output group can contain only one video JPEG encode. Put that single encode in one output.

Organize encodes in an HLS or MediaPackage output group

An HLS or MediaPackage output group is typically set up as a video ABR stack. The term *ABR* stands for *adaptive bitrate*. A video *ABR stack* is an output group that contains the following:

- Multiple versions (renditions) of the video. Each rendition has a different resolution.
- One or more audio encodes.
- One or more captions encodes.

There are two ways to organize the encodes, depending on whether the audio encodes must be bundled or each in their own rendition. You should have already obtained this information from your downstream system.

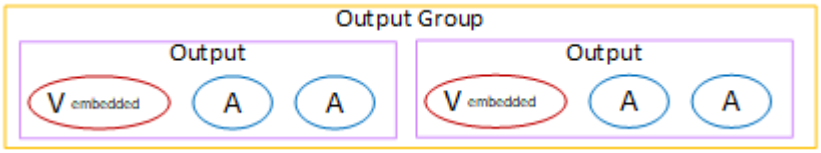
Downstream players that require bundled audio

Plan for the output group to contain the following:

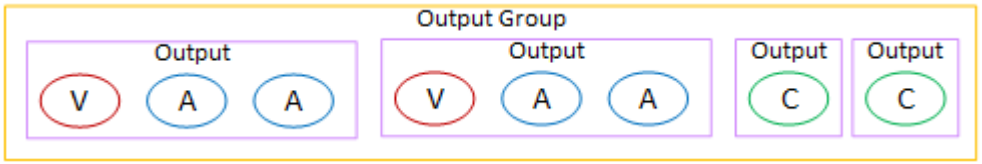
- One output for each video encode. This output holds one video encode, all the audio encodes, and all the captions encodes (if the captions are embedded).

  The same audio encodes will appear in each output. For example, the English and French encodes will appear in the high-resolution output, then the same English and French encodes will appear in the low-resolution output.
- One output for each captions encode, if the captions are sidecars.

This diagram illustrates an HLS output group when the captions encodes are embedded.



This diagram illustrates an HLS output group when the captions encodes are sidecars.



Downstream players that require separate audio

Plan for the output group to contain the following:

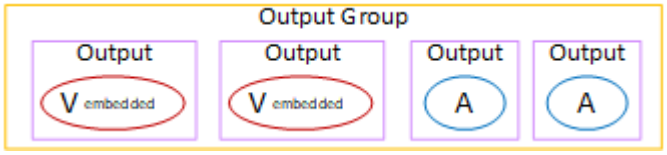
- One output for each video encode. This output holds one video and all the captions encodes (if the captions are embedded).

- One output for each audio encode.

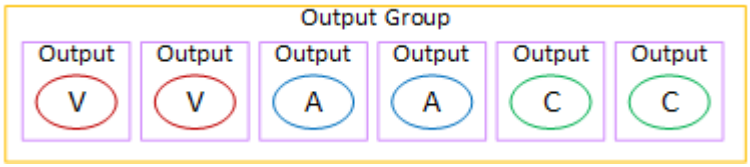
  The audio encodes might be for different languages, or they might be for different bitrates, or they might be for different languages and bitrates.
- One output for each captions encode, if the captions are sidecars.

The arrangement of the audio encodes in this output group is called an *audio rendition group*.

This diagram illustrates an HLS output group with an audio rendition group, and with embedded captions encodes.



This diagram illustrates an HLS output group for an ABR stack with an audio rendition group, and with sidecar captions encodes.



Organize encodes in a Microsoft Smooth output group

A Microsoft Smooth output group is typically set up as a video ABR stack. The term *ABR* stands for *adaptive bitrate*. A video *ABR stack* is an output group that contains the following:

- Multiple versions (renditions) of the video. Each rendition has a different resolution.
- One or more audio encodes.
- One or more captions encodes.

There are two ways to organize the encodes, depending on whether the audio encodes must be bundled or each in their own rendition. You should have already obtained this information from your downstream system.

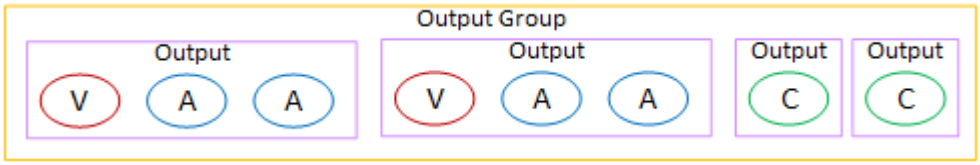
Downstream players that require bundled audio

Plan for the output group to contain the following:

- One output for each video encode. This output holds one video encode and all the audio encodes.

  The same audio encodes will appear in each output. For example, the English and French encodes will appear in the high-resolution output, then the same English and French encodes will appear in the low-resolution output.
- One output for each captions encode. The captions in a Microsoft Smooth output group are always sidecar captions.

This diagram illustrates a Microsoft Smooth output group with bundled audio.



Downstream players that require separate audio

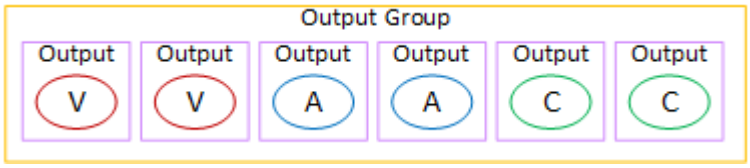
Plan for the output group to contain the following:

- One output for each video encode.
- One output for each audio encode.

  The audio encodes might be for different languages, or they might be for different bitrates, or they might be for different languages and bitrates.
- One output for each captions encode. The captions in a Microsoft Smooth output group are always sidecar captions.

The arrangement of the audio encodes in this output group is called an *audio rendition group*.

This diagram illustrates a Microsoft Smooth output group with an audio rendition group.

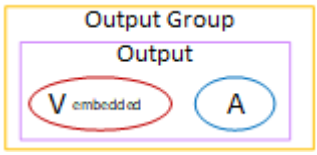


Organize encodes in an RTMP output group

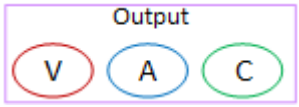
An RTMP output group contains one video encode, one audio encode, and one captions encode.

Plan for the output group to contain one output that holds all the encodes.

This diagram illustrates an RTMP output group where the captions are embedded in the video encode.



This diagram illustrates an RTMP output group with object-style captions.

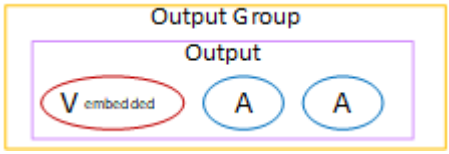


Organize encodes in a UDP output group

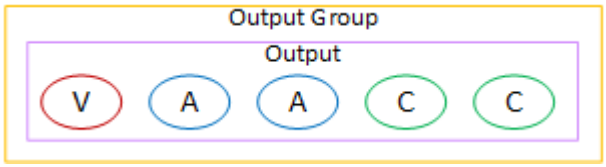
A UDP output group can contain one video encode, one or more audio encodes, and one or more captions encodes (either embedded or object-style).

Plan for the output group to contain one output that holds the single video encode, all the audio encodes, and all the captions encodes.

This diagram illustrates a UDP output group where the captions are embedded in the video encode.



This diagram illustrates a UDP output group with object-style captions.



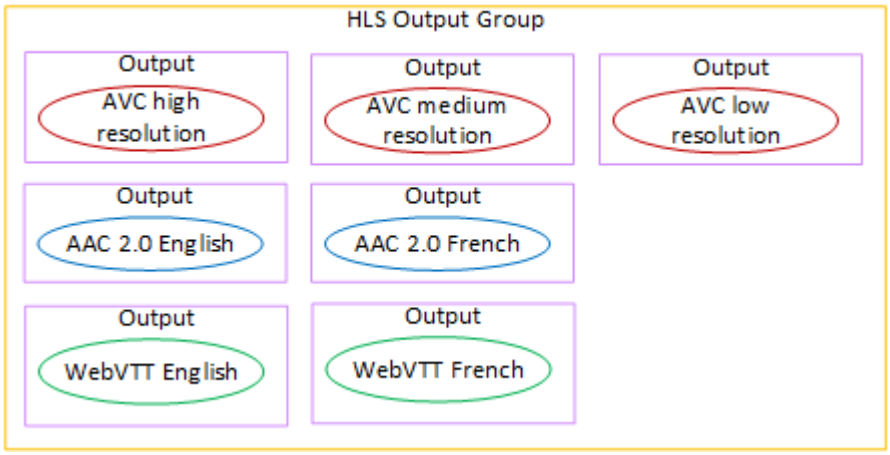
Examples

the section called “Example of a plan” shows an example of a workflow that includes three output groups. The table in this section shows the encodes that you might include in each output group.

This section shows the results of organizing the encodes in those output groups.

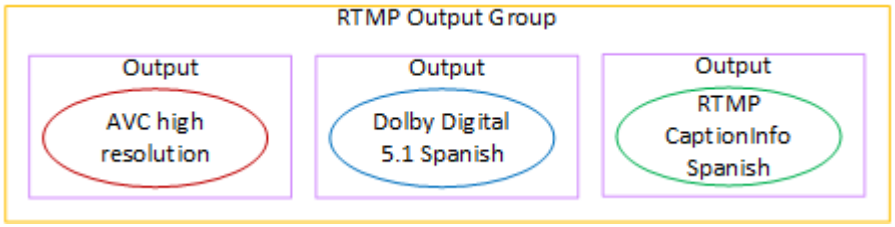
HLS output group

The example of an HLS output group contains three videos, each with a different resolution. The audio encodes are each in their own output, which means that the output group contains an audio rendition group. The captions are WebVTT, which is a sidecar style of captions. Therefore, each captions encode goes in its own output.



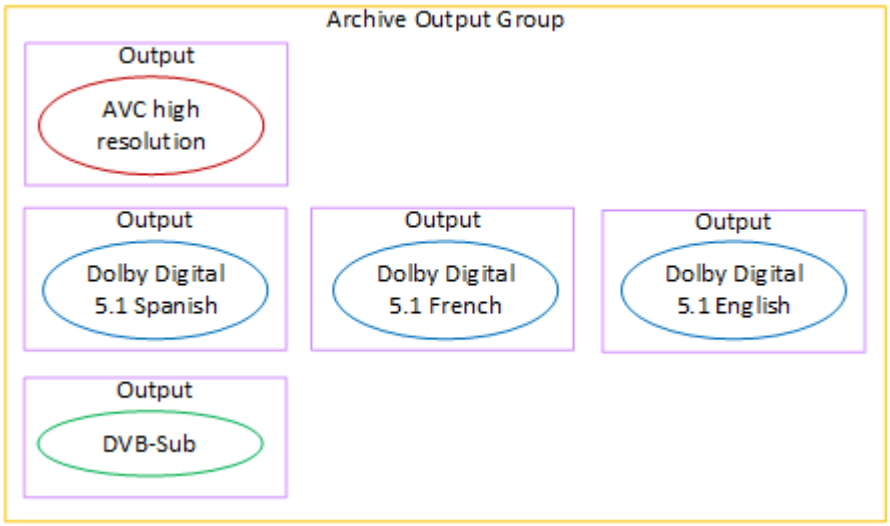
RTMP output group

The example of an RTMP output group contains one video, one audio, and one captions encode.



Archive output group

The example of an Archive output group contains one video encode, three audio encodes, and one captions encode. In an Archive output, the video and audio encodes are always each in their own output. In this example, the captions are WebVTT, which is a sidecar style of captions. Therefore, each captions encode goes in its own output.



Step 4: Design the encodes

In the first step of planning the channel, you identified the video, audio, and captions encodes to include in each output group. In the third step, you organized these encodes into outputs in each output group.

You must now plan the configuration parameters for each encode. As part of this plan, you identify opportunities for sharing encodes among outputs in the same output group in the channel, and among outputs in different output groups in the channel.

Result of this procedure

After you have performed this procedure, you will have a list of video, audio, and captions encodes to create.

Topics

- Plan the encodes
- Identify encode sharing opportunities

Plan the encodes

In the section called “Step 2: Map outputs to sources”, you sketched out a plan for the encodes you want to create in each output group. Below is the example of the plan from that step, showing the outputs and encodes, and the sources for those encodes.

At some point, you must fill in the details for the encodes identified in the second and third columns of this table. You have a choice:

- You can decide these details now.

- You can decide the details later, when you are actually creating the channel. If you decide to do this, we recommend you still read the procedures after the table, to get an idea of what is involved in defining an encode.

Example

Output group	Type of encode	Encode nickname	Characteristics of the encode	Source	Identifier in source
HLS	Video	VideoA	AVC 1920x1080, 5 Mbps	HEVC	PID 600
		VideoB	AVC 1280x720, 3 Mbps	HEVC	PID 600
		VideoC	AVC 320x240, 750 Kbps	HEVC	PID 600
	Audio	AudioA	AAC 2.0 in English at 192000 bps	AAC 2.0	PID 759
		AudioB	AAC 2.0 in French at 192000 bps	AAC 2.0	PID 747
	Captions	CaptionsA	WebVTT (object-style) converted	Embedded	Channel 4

Output group	Type of encode	Encode nickname	Characteristics of the encode	Source	Identifier in source
			from embedded, in English		
		CaptionsB	WebVTT (object-style) converted from embedded, in French	Embedded	Channel 2
RTMP	Video	VideoD	AVC 1920x1080, 5Mbps	HEVC	PID 600
	Audio	AudioC	Dolby Digital 5.1 in Spanish	Dolby Digital 5.1	PID 720
	Captions	CaptionsC	RTMP CaptionInfo (converted from embedded) in Spanish	Embedded	Channel 2
Archive	Video	VideoE	AVC, 1920x1080, 5 Mbps	HEVC	PID 600
	Audio	AudioD	Dolby Digital 2.0 in Spanish	AAC 2.0	PID 746

Output group	Type of encode	Encode nickname	Characteristics of the encode	Source	Identifier in source
		AudioE	Dolby Digital 2.0 in French	AAC 2.0	PID 747
		AudioF	Dolby Digital 2.0 in English	AAC 2.0	PID 759
	Captions	CaptionsD	DVB-Sub (object-style) converted from Teletext, in 6 languages.	Teletext	PID 815

Design the details for each video encode

For each video encode in your table, you have already identified the source asset, codec, resolution and bitrate. You must now identify all the other encoding parameters you need to set.

Follow this procedure for each individual video encode.

1. Look at the fields in the video encode section of each output. To view these fields, follow these steps. Don't worry about not completing all the sections. You only want to display the video encode fields, and you will then cancel the channel.

   - On the MediaLive home page, choose **Create channel**, and in the navigation pane, choose **Channels**.

     If you've created a channel before, you won't see the home page. In that case, in the MediaLive navigation pane, choose **Channels**, and then choose **Create channel**.
   - On the **Create channel** page, under **Output groups**, choose **Add**.

     Don't worry that you haven't completed any of the earliers sections in the channel. You are only trying to display all the fields for the video encode.
   - In the **Add output group** section, choose **HLS** and choose **Confirm**.

- Under that output group, choose **Output 1**.
- In the **Output** section, go to the **Stream settings** section, and choose the **Video** link.
- In the **Codec settings** field, choose the codec that you want for this video encode. More fields appear. Choose the field labels for all the sections to display all the fields.

2. In each section, determine whether you need to change the defaults.

   - Many of the fields have defaults, which means you can leave the field value as is. For details about a field and its default value, choose the **Info** link next to the field.
   - There are some fields that you might need to set according to instructions from your downstream system, to match the expectations of the downstream system.
   - There are some fields where the value you enter affects the output charges for this channel. These are:
     - The **Width** and **Height** fields (which define the video resolution).
     - The **Framerate** fields.
     - The **Rate control** fields.

     For information about charges, see the MediaLive price list.
   - You can read about some of the fields in the following sections:
     - For information about the **Color space** fields, see the section called "Video – complex color space conversion".
     - For information about the Additional encoding settings fields, see the section called "Video – enhanced VQ"
     - For information about the **Rate control** fields, see the section called "Video – rate control mode". There are fields in this section that affect the output charges for this channel. For more information about charges, see the MediaLive price list.
     - For information about the **Timecode** fields, see the section called "Timecodes".

3. Make detailed notes about the values for all the fields you plan to change. Do this for every video encode that you identified.

Design the details for each audio encode

For each audio encode in your table, you have already identified the source asset, codec and bitrate. You must now identify all the other encoding parameters you need to set.

Follow this procedure for each individual audio encode.

1. Look at the fields in the audio encode section of each output. To view these fields, follow the same steps as for the video encodes, but choose the **Audio 1** link.

   With audio encodes, there aren't many fields for each code. But the fields for the codecs are very different from each other.
2. Study the fields and make notes.

Design the details for each captions encode

For each captions encode in your table, you have already identified the source captions, format, and language. You must now identify all the other encoding parameters you need to set.

Follow this procedure for each individual captions encode.

1. Look at the fields in the captions encode section of each output. To view these fields, follow the same steps as for the video encodes, but choose Add caption to add a captions section, because there is no captions section by default.

   With captions encodes, there aren't many fields for each captions format. But the fields for the formats are very different from each other.
2. Study the fields and make notes.

Identify encode sharing opportunities

If you have already identified the details for all the output encodes, you can now identify opportunities for encode sharing.

If you plan to identify details later, we recommend that you come back to this section to identify opportunties.

Read about encode sharing and encode cloning in the section called “Sharing and cloning encodes”.

You will use encode sharing and encode cloning when you create the encodes in the channel, starting with the section called “Set up video”.

- When you have a complete list, compare the values for the encodes:
  - If you have two (or more) encodes with identical values, you can share the encode. When you create the channel, you can create this encode once, in one output. You can then reuse

that encode in other outputs. The procedure for creating the encode provides detailed instructions for reusing.

Keep in mind that two encodes are identical only if they are identical in all their fields, including sharing the same video source. For example, in the sample table earlier in this section, the first video encode for HLS and the video encode for RTMP share the same video source.

- If you have two (or more) encodes with nearly identical values, you can clone an encode to create a second encode, and then change specific fields in the second encode. The procedure for creating the encode provides detailed instructions for cloning.

Then identify opportunities for sharing, in the same way as you did for the video encodes. Keep in mind that two encodes are identical only if they are identical in all their fields, including sharing the same audio source.

Carefully identify the video encodes to share by noting the outputs and output groups each belongs to.

Then identify opportunities for sharing, in the same way as you did for the video encodes. Keep in mind that two encodes are identical only if they are identical in all their fields, including sharing the same captions source.

Example

Following from the example in the earlier steps in this section about channel planning, you might decide you have these opportunities shown in the last two columns of this table.

Encode nickname	Characteristics of the encode	Source	Opportunity	Action
VideoA	AVC 1920x1080, 5 Mbps	HEVC		Create this encode from scratch.
VideoB	AVC 1280x720, 3 Mbps	HEVC	Clone	Clone VideoA and change the

Encode nickname	Characteristics of the encode	Source	Opportunity	Action
				bitrate. Perhaps also other fields.
VideoC	AVC 320x240, 750 Kbps	HEVC	Clone	Clone VideoA and change the bitrate and perhaps other fields.
AudioA	AAC 2.0 in English at 192000 bps	AAC 2.0		Create this encode from scratch.
AudioB	AAC 2.0 in French at 192000 bps	AAC 2.0	Clone	Clone AudioA and change the audio selector (the reference to the source) to the selector for French. Perhaps also change other fields.
CaptionsA	WebVTT (object-style) converted from embedded, in English	Embedded		Create this encode from scratch.

Encode nickname	Characteristics of the encode	Source	Opportunity	Action
CaptionsB	WebVTT (object-style) converted from embedded, in French	Embedded	Clone	Clone CaptionsC and change the captions selector (the reference to the source) to the selector for French. Perhaps also change other fields.
VideoD	AVC 1920x1080, 5Mbps	HEVC	Share	Share VideoA
AudioC	Dolby Digital 5.1 in Spanish	Dolby Digital 5.1		Create this encode from scratch.
CaptionsC	RTMP CaptionInfo (converted from embedded) in Spanish	Embedded	Clone	Clone CaptionsA and change the captions selector (the reference to the source) to the selector for Spanish. Perhaps also change other fields.
VideoE	AVC, 1920x1080, 5 Mbps	HEVC	Share	Share VideoA

Encode nickname	Characteristics of the encode	Source	Opportunity	Action
AudioD	Dolby Digital 2.0 in Spanish	AAC 2.0		Create this encode from scratach. Although its source is the same as Aa, its output codec is different, which means all its configura tion fields are different. Therefore, there is no advantage to cloning.

Encode nickname	Characteristics of the encode	Source	Opportunity	Action
AudioE	Dolby Digital 2.0 in French	AAC 2.0	Clone	Clone AudioD and change the audio selector (the reference to the source) to the selector for French. Perhaps also change other fields. Don't clone AuduioB because AudioB and AudioA have different output codecs. Therefore, there is no advantage to cloning.

Encode nickname	Characteristics of the encode	Source	Opportunity	Action
AudioF	Dolby Digital 2.0 in English	AAC 2.0	Clone	Clone AuduioD and change the audio selector (the reference to the source) to the selector for English. Perhaps also change other fields. Don't clone AudioB because AudioB and AudioF have different output codecs. Therefore, there is no advantage to cloning.
CaptionsD	DVB-Sub (object-style) converted from Teletext, in 6 languages.	Teletext		Create this encode from scratch.

Next steps

You have now planned the channel:

- You have mapped your sources to your outputs. On the output side, you have identified the video, audio and captions *encodes* that you want. And on the source side, you have identified the specific video, audio, and captions *assets* that can produce those encodes.

- You have then organized these encodes into outputs within the output groups that you identified when you planned your workflow.
- You have also identified the opportunities for sharing assets among different outputs within an output group or among different output groups.

You are now ready to create the inputs that you identified when you assessed the source formats. To create inputs, see *Setup: Creating inputs*.

Setup: Creating inputs

This section describes how to create inputs for the content sources for a MediaLivechannel. You must create these inputs before you start to create the channel.

To create an input, you must perform these steps:

- You must arrange for the operator at the upstream system to perform some setup.
- You must create inputs in MediaLive.

These two steps create a connection between an address on the upstream system and an address on AWS Elemental MediaLive. The source content moves from the specified address on the upstream system to the specified address on MediaLive as either a *push* by the upstream system or a *pull* by MediaLive. The connection information is contained in the input that you create.

The setup you perform is different for each combination of upstream system (format and delivery protocol) and input type. If you haven't already done so, you must identify the upstream system and input type for each content source. See the section called “Step 4: Assess the upstream system ”.

Topics

- Next steps

Getting ready

Before you create any input, you should plan the workflow. Read the following sections:

- *Setup: Preparing upstream and downstream* – You must set up for delivery from the upstream system. The task of creating an input is part of that delivery setup. Before you can create the input, you must coordinate with your upstream system and content provider.
- the section called "Pipeline redundancy" – You must decide if you want to implement pipeline redundancy—whether you set up a standard channel or a single-pipeline channel. Implementing pipeline redundancy provides resiliency in the channel processing pipeline.
- the section called "Automatic input failover" – You must decide if you want to implement automatic input failover. Implementing automatic input failover provides resiliency upstream of the channel, for one of the channel's inputs.

Setting up a CDI input

This section describes how to create a CDI push input. Create the input before you create the channel that ingests the input.

> **Note**
>
> Make sure that the content provider is using the latest version of the AWS CDI SDK on their CDI source device.

With a CDI source, the upstream system *pushes* the content to MediaLive.

To perform this setup, you must work with an Amazon VPC user, with an operator at the upstream system, and you must work within MediaLive.

Topics

- Step 1: Request setup on the VPC
- Step 2: Create a CDI input
- Step 3: Ensure correct setup on the upstream system

- Result of this procedure

Step 1: Request setup on the VPC

An Amazon VPC user must set up the VPC, and identify subnets and security groups that both the upstream system and MediaLive will use.

To set up the VPC

1. Provide the Amazon VPC user with the following guidelines.

   - Guideline for the subnets – Request two subnets. You need two subnets because a CDI input is always a standard-class input, even if your channel is a single-pipeline channel. For information about input classes, see the section called "Channel classes and input classes".

     These rules apply:

     - The two subnets must be in different Availability Zones.
     - Each subnet must have a private CIDR block (a range of IP addresses).
     - Each subnet must have at least two unused addresses in that block—one for the upstream system and one for the CDI input.
     - Any other VPC-based sources (source B) that you create for use in the same channel as this CDI source (source A) must be in subnets that are in the same Availability Zones as source A. The two subnets of the source B can be different from the source A, but the Availability Zones of those two subnets must be the same as the Availability Zones of source A.
   - Guideline for the security group – the security groups or groups for each subnet must follow these rules:
     - The combined inbound rules of the security groups must allow inbound traffic from the IP addresses of the upstream system that is in that subnet.
     - The subnet must have an EFA-enabled security group. To create this type of security group and for information about its rules, see the Amazon Elastic Compute Cloud User Guide.
2. After the Amazon VPC user has performed the setup, obtain the following information:

   - The ID of the VPC. For example: `vpc-3f139646`
   - The IDs of the two subnets. For example, one subnet might have this ID: `subnet-1122aabb`
   - The IDs of the security groups for the subnet or subnets. For example: `sg-51530134`

Step 2: Create a CDI input

After the Amazon VPC user has set up on the VPC, you can create the CDI input in MediaLive.

This section describes how to create a regular CDI input. Create this type of input if you don't plan to support automatic input failover for the CDI source attached to the channel. (If you do plan to implement it, create CDI *partner inputs* instead.)

To create a CDI push input

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Inputs**. On the **Inputs** page, choose **Create input**.
3. Complete the **Input details** section:
   - **Input** name – enter a name.
   - **Input type** – choose **AWS CDI**.
4. Complete the **VPC settings** section:
   - Choose **Select subnets and security groups**.
   - For **Subnets**, choose one of the subnets that you obtained. The dropdown list shows subnets in all VPCs, identified as follows:

     ```
     <subnet ID> <Availability Zone of subnet> <IPv4 CIDR block of subnet>
     <VPC ID> <Subnet tag called "Name", if it exists>
     ```

     For example:

     subnet-1122aabb us-west-2a 10.30.30.0/24 vpc-3f139646 Subnet for MLive push inputs

     If the list of subnets is empty, choose **Specify custom VPC**, and enter the subnet ID in the field. (You need to enter only the subnet ID, for example, **`subnet-1122aabb`**.)
   - In **Subnets**, choose the second subnet. This second time, the dropdown list shows only the subnets in the same VPC as the first subnet.
   - For **Security groups**, choose the security group or groups that you obtained, following the same process as for the subnets. The dropdown list shows security groups belonging to the VPC that you chose, identified as follows:

     ```
     <security group ID> <description attached to this security group> <VPC
     ID>
     ```

For example:

sg-51530134 Security group for MLive push inputs vpc-3f139646

5. Complete the **Role ARN** section to choose a role for MediaLive to use with this input. For more information, see the section called “IAM role and ARN”.
6. In the **Tags** section, create tags if you want to associate tags with this input. For more information, see the section called “Tagging resources”.
7. Choose **Create**.

   MediaLive creates the input and automatically creates two endpoints on that input. These endpoints have a private IP address from the subnet range, and they specify port 5000. For example:

   ```
   10.30.30.33:5000
   ```

   ```
   10.30.30.44:5000
   ```
8. Provide the upstream system with these endpoints:

   - If you will set up the channel as a standard channel, provide both endpoints. The upstream system must push the content to both endpoints.
   - If you will set up the channel as a single-pipeline channel, provide only the first endpoint. The upstream system must push to this one endpoint.

IAM role and ARN

This section describes how to complete the **Role ARN** section on the **Create input** pane of the MediaLive console.

You must choose a role for MediaLive to assume when it creates an RTP Push input. To create the input, MediaLive must obtain the network interfaces for the two endpoints in the input. These endpoints are in the CIDR range of the subnets that you identified. As soon as you choose **Create** for this input, MediaLive requests these network interfaces from Amazon VPC. The role that you choose ensures that MediaLive succeeds in its request to Amazon VPC.

> **Note**
>
> This section on the MediaLive console is identical to the **IAM role** section on the **Create channel** page (also on the MediaLive console). The difference in the two usages is that on the **Create input** page, you are attaching the role to the input. On the **Create channel** page, you are attaching the role to the channel. You can use the same role (for example, the **MediaLiveAccessRole**) in both usages.

There are two general scenarios for choosing a role, depending on whether your organization has a designated administrator.

Your organization has a designated administrator

Your organization might have an administrator who manages this service. That administrator has likely set up one or more roles:

- Ask the administrator or your manager which role to use. Or if only one role is listed in **Use existing role**, choose that role.
- If the only role that is listed is **MediaLiveAccessRole**, choose that role. In addition, if the **Update** button is displayed beside this role name, choose the button. (The button does not always appear, but whenever it does appear, choose it to refresh the role.)
- If you want the selected role to appear first in the list next time, select **Remember ARN**.

Your organization has no administrator

Your organization might not have a designated service administrator. In this case, if none of your colleagues have set up a suitable role, you might have to create one yourself and then choose it.

- You can create the default role, called **MediaLiveAccessRole**. To first check if someone else has already created this role (only one person needs to create it for all users in your AWS account), look at **Create role from template**:
  - If this option is grayed out, this task has been done. In that case, choose **Use existing role**, and then choose **MediaLiveAccessRole** from the list.
  - If this option is not grayed out, choose **Create role from template**, and then choose **Create IAM role**. Next, choose that role from the list. If MediaLive does not let you create the role, speak to an administrator in your organization about your IAM permissions.

- If the **MediaLiveAccessRole** has already been created and the **Update** button is displayed beside it, choose the button. (The button does not always appear, but whenever it does appear, choose it to refresh the role.)
- If you want the selected role to appear first in the list next time, select **Remember ARN**.

Step 3: Ensure correct setup on the upstream system

After you create the CDI input, you must make sure that the operator at the upstream system sets up correctly with your VPC, and that they push content to the correct locations in MediaLive.

To set up for a standard channel

If the planned channel is a standard channel, you must ensure that the operator at the upstream system provides two sources.

1. Provide the operator with this information:

   - The IDs of the VPC, two subnets, and the security groups that the Amazon VPC user gave you in step 1.
   - The two endpoints (URLs) that MediaLive generated when you created the CDI input. These endpoints are the addresses in the blue boxes in the diagram after this procedure. These URLs each have a private IP address from the subnet range, and they specify port 5000. For example:

     ```
     10.30.30.33:5000
     ```

     ```
     10.40.40.44:5000
     ```

2. Make sure that the operator sets up properly for a standard channel. They must do the following:

   - Set up two output interfaces. Set up one upstream system with one output interface in one of the subnets, and set up the other upstream system with one output interface in the other subnet. These interfaces are the addresses in the purple boxes in the diagram after this procedure.
   - Make sure that the two content sources are identical in terms of video resolution and bitrate.
   - Push to the correct URLs on MediaLive. For example, they must push to:

     ```
     10.30.30.33:5000
     ```

```
10.40.40.44:5000
```

To set up for a single-pipeline channel

- There will be one upstream system that sends content to only one of the subnets in the VPC.
- The content will flow from the VPC to one of the endpoints on the input. The other endpoint will never be used.
- MediaLive will ingest the single source content.

1. Provide the operator with this information:

   - The IDs of the VPC, one of the subnets, and all of the security groups that the Amazon VPC user gave you.
   - Only the first of the two endpoints (URLs) that MediaLive generated when you created the CDI input. These endpoints are the addresses in the blue box in the diagram after this procedure. The URL has a private IP address from the subnet range, and it specifies port 5000.

     ```
     10.30.30.33:5000
     ```

2. Make sure that the operator sets up properly for a single-pipeline channel. They must:

   - Set up one upstream system.
   - Set up one output interfaces. The interface is the address in one of the purple boxes in the diagram after this procedure.
   - Push to the correct URL on MediaLive. For example, they must push to:

     ```
     10.30.30.33:5000
     ```

Result of this procedure

The results of this setup are illustrated in the diagram that follows. There are three main components:

- The upstream system (purple boxes).
- The VPC, with subnets (green boxes), and VPC security groups (yellow boxes).

- The CDI input (blue box).

The CDI input has one or two *endpoint* URLs (the addresses in the blue box). These endpoints are elastic network interfaces (ENIs) on your VPC. MediaLive has permission to use these ENIs for its inputs. MediaLive has permission (through the IAM trusted entity role) to automatically manage the ENIs for its inputs.

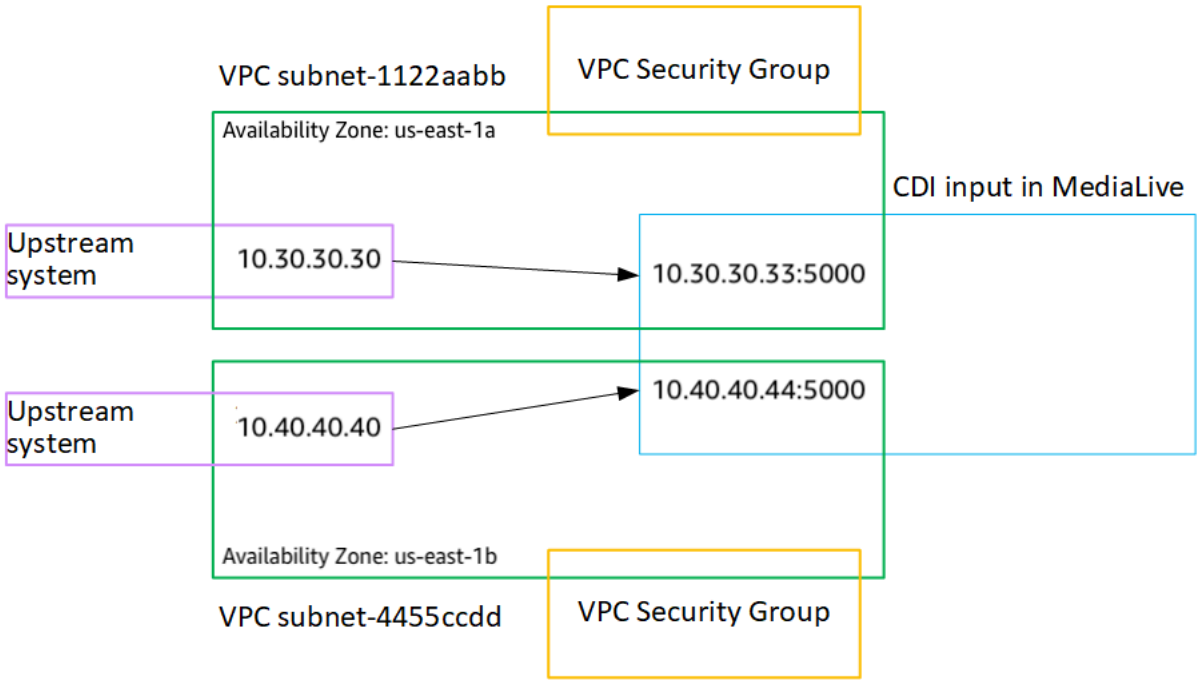
The upstream system has two outputs. Each output has an IP address in one of the specified subnets in your VPC. The upstream system has permission (through the rules in one or more Amazon VPC security groups) to push content to these endpoints. The upstream system pushes the source content to both endpoints (if you are setting up a standard channel) or to one endpoint (if you are setting up a single-pipeline channel).

The upstream system has IP addresses in the VPC subnets, and the CDI input has endpoints in the same VPC subnets. In this way, the delivery of the content from the upstream system to MediaLive takes place within the security of the VPC.

The two IP addresses on the CDI input are fixed for the lifetime of the input. They are fixed, regardless of changes such as modifying other information in the input, or attaching the input to a different channel.

Keep in mind that with a push input, the upstream system must be pushing the video source to the input when you start the channel. The upstream system does not need to be pushing before then.

At runtime of the channel, MediaLive reacts to the content that is being pushed and ingests it.



Creating a partner CDI push input in Amazon VPC

A partner CDI input is a specific configuration of a CDI input. If you want to support automatic input failover for the CDI source attached to the channel, you must set up the two CDI inputs as *partners*. For more information about partner CDI inputs, see the section called "CDI inputs as partner inputs".

The two inputs always work together, as the two inputs in an automatic failover pair. The two inputs can be used only together, as a failover pair.

You create a set of partner CDI inputs in two steps:

- Create the first partner CDI input in the usual way.
- Then, create the second partner input from the first input.

To create the first partner CDI input

- If you already have a regular CDI input, you can use it as the first partner. Skip this step and go to the step for creating the second partner, below.

  If not, create the input in the usual way.

MediaLive creates the input and automatically creates two endpoints on that input. These endpoints each have a private IP address from the subnet range, and they specify port 5000. For example:

```
10.30.30.33:5000
```

```
10.30.30.44:5000
```

Don't provide this information to the upstream system until you have created the second partner.

To create the second partner CDI input

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Inputs**.
3. On the list of inputs, choose the first partner input. The details for the input appear.

   In the **Endpoints** section, you can see the endpoints that apply to this input. For example:

   ```
   10.30.30.33:5000
   ```

   ```
   10.30.30.44:5000
   ```
4. At the top of the page, choose **Create partner input**.
5. On the confirmation dialog, optionally choose to copy the tags, if any, from the first input.
6. Choose **Confirm**.

   The **Input details** page for this input appears, showing information about the new input.

   - In **Details**, the **Name** shows that the input has the same name as the first input, with the suffix "- partner".
   - In **Details**, the partner **CDI ID** field shows the ID of the first input.
   - In **Endpoints**, the endpoints for the input are identical to the two endpoints for the first input, except that the port numbers are different. For example:

     ```
     10.30.30.33:5001
     ```

     ```
     10.30.30.44:5001
     ```

Setting up an Elemental Link input

This section describes how to create an Elemental Link push input. Create the input before you create the channel that ingests the input.

The AWS Elemental Link device *pushes* the content to MediaLive.

To perform this setup, you must work with an operator of the AWS Elemental Link device.

Topics

Step 1: Obtain information

Obtain the following information from the operator of the AWS Elemental Link device:

- The name of the device or devices that will provide your source. For example:

  hd-re87jr7crey

  You need two device names for a standard-class input, or one device name for a single-class input. For information about input classes and their uses, see the section called "Channel classes and input classes".
- The Region that the device is configured for, so that you can set MediaLive for that Region. These rules apply:
  - Both devices must be in the same Region.
  - The device, the input for that device, and the channel that uses the input must all be in the same Region.

Step 2: Create an Elemental Link input

After you have obtained information about the AWS Elemental Link hardware device, you can create an Elemental Link input.

To create a Link input

1. Make sure that you have the information from step 1.
2. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
3. Set the AWS Region to match the Region where the AWS Elemental Link device exists.
4. In the navigation pane, choose **Inputs**. On the **Inputs** page, choose **Create input**.
5. Complete the **Input details** section:
   - **Input** name – enter a name.
   - **Input type** – choose **Elemental Link**.
6. In the **Input devices** section, for **Input class**, choose the class for this input:
   - STANDARD_INPUT
   - SINGLE_INPUT
7. In **Input devices**, choose one or two devices to attach to this input as the source. From the dropdown lists, choose the device names you previously obtained. The lists show only the devices that are set up in the current Region.
   - If the input is a standard-class input, complete both fields, to provide two source devices.
   - If the input is a single-class input, complete the first field and leave the second field empty.
8. In the **Tags** section, create tags if you want to associate tags with this input. For more information, see the section called "Tagging resources".
9. Choose **Create**.

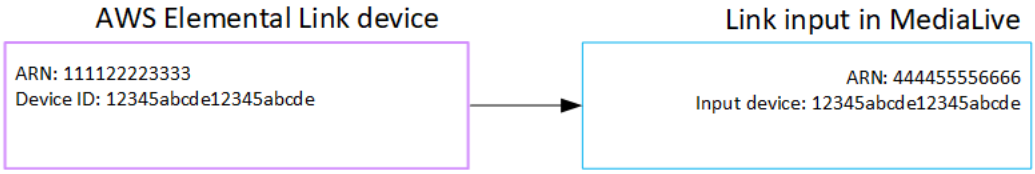
   The **Details** pane appears for the input, showing details about the input and the MediaLive device that it uses, including the following:

   - **ID** – A unique numerical ID for the input.
   - **ARN** – An input ARN that includes that numerical ID.
   - **Input device** – The unique ID of the AWS Elemental Link device.
   - **Device thumbnail** – A thumbnail of the content that is currently being pushed by the device, if there is any being pushed. The device generates the thumbnails by capturing a video frame approximately every 5 seconds.

Result of this procedure

As a result of this setup, an Elemental Link input (the blue box) exists that identifies the AWS Elemental Link device or devices (the purple box) that are connected to MediaLive. There is no other setup for you to perform, because the AWS Elemental Link device is designed to work seamlessly with MediaLive.

At runtime of the channel, MediaLive reacts to and ingests the content that AWS Elemental Link is pushing.



Setting up an HLS input

This section describes how to create an HLS input. Create the input before you create the channel that ingests the input.

With an HLS input, MediaLive connects to the upstream system when the channel starts and *pulls* the sources.

To perform this setup, you must work with an operator at the upstream system.

Topics

Step 1: Obtain information

Obtain the following information from the operator at the upstream system:

- The locations (URLs) on the upstream server where the M3U8 manifest files are stored.

There are two URLs for a standard-class input, or one URL for a single-class input. For information about input classes and their uses, see the section called "Channel classes and input classes".

See the tables later in this section for the URL format and for examples.

Make a note of the full URLs.

- The user name and password (credentials) to access the upstream server, if the upstream system requires authenticated requests, and to access the license server, if the HLS source is encrypted. You might need credentials for the upstream system, or for the license server, or both.

  If you need credentials for both, the credentials must be identical for both servers. When you discussed any encryption requirements with the upstream system, you should have made sure that the license server uses the same credentials as the upstream system.

  Note that these user credentials relate to user authentication, not to the protocol. User authentication is about whether the upstream system or license server will accept your request. The protocol is about whether the request is sent over a secure connection.

Upstream server is an HTTP or HTTPS server

Format of URL	`http//:<web server>[:port]/<path>/<file>.m3u8`
	
	or
	`https//:<web server>[:port]/<path>/<file>.m3u8`

Example	`https://203.0.113.13/sports/curling.m3u8` and `https://198.51.100.54/sports/curling.m3u8`

Upstream server is AWS Elemental MediaStore

Format of URL	**mediastoressl://<data endpoint for container>/<path>/<file>.m3u8**
Example	Assume that the data endpoint for the container for one of the content sources is the following: **eri39n.data.mediastore.us-west-2.amazonaws.com** . Assume that the M3U8 file is called `curling.m3u8` , and it is stored in the container, in the path `sports/canada` . The URL for one of the content sources would be: **mediastoressl://eri39n.data.mediastore.us-west-2.amazonaws.com/sports/canada/curling.m3u8** .

Upstream server is Amazon S3

Upstream server	Format of URL
Format of URL	`s3ssl://<bucket>/<path>/<file>.m3u8`
Example	`s3ssl://DOC-EXAMPLE-BUCKET/movies/main/mlaw.m3u8` and `s3ssl://DOC-EXAMPLE-BUCKET1/movies/redundant/mlaw.m3u8`

Step 2: Create an HLS input

After you have obtained information from the upstream system, you can create an HLS input.

To create an HLS pull input

1. Make sure that you have the information from step 1.
2. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
3. In the navigation pane, choose **Inputs**. On the **Inputs** page, choose **Create input**.
4. Complete the **Input details** section:
   - **Input** name – enter a name.
   - **Input type** – choose **HLS**.
5. In the **Input class** section, choose the class for this input:
   - STANDARD_INPUT
   - SINGLE_INPUT
6. In the **Input sources** section, enter the URLs you previously obtained:
   - If the input is a standard-class input, complete both fields, to provide two URLs.
   - If the input is a single-class input, complete the first field with the URL that you obtained and leave the second field empty.
7. If the upstream system and/or the license server (if the HLS source is encrypted) requires that you provide user credentials, you must also enter the user name and password key for accessing the location. These credentials are stored on the Systems Manager Parameter Store. For more information, see the section called “About the feature for creating password parameters”.

   If one of the servers (upstream system or license server) requires credentials and the other doesn't, MediaLive presents them to both. But the server that doesn't need them simply ignores them.
8. In the **Tags** section, create tags if you want to associate tags with this input. For more information, see the section called “Tagging resources”.
9. Choose **Create**.

MediaLive creates the input and adds it to the list of inputs. The input specifies either one or two sources. The sources don't appear in the list, but if you choose the **Name** link, the details page shows them.

Step 3: Ensure correct setup on the HLS upstream server

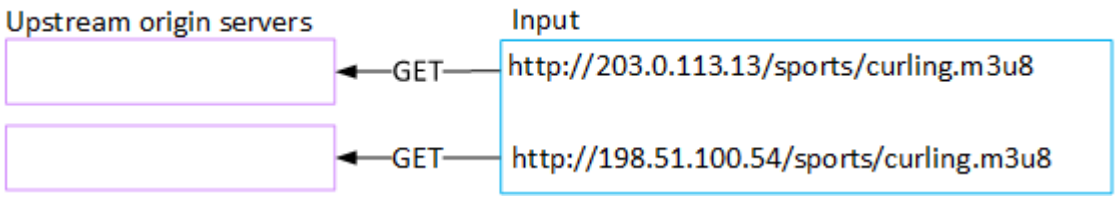
An operator at the upstream server must set up the source content on the upstream system. Make sure that the operator sets up as follows:

- They set up to deliver the correct number of sources:
  - If the MediaLive channel is a standard channel, the operator must set up two sources for the content. They must make sure that the two sources are identical in terms of video resolution and bitrate.
  - If the MediaLive channel is a single-pipeline channel, the operator must set up one source for the content.
- They set up to make the M3U8 manifest files available at the agreed URLs. These are the URLs that you obtained in step 1, and that you configured into the HLS input. They correspond to the URLs shown in the diagram after this procedure.

Result of this procedure

As a result of this setup, an HLS input exists that specifies one or two *source* URLs. These sources are the URLs for the source content on the upstream server. When you start the channel, MediaLive will connect to the upstream system at this source location or locations and pull the HLS manifests into MediaLive:

- For a channel set up as a standard channel, MediaLive expects the upstream system to provide two sources and will therefore attempt to pull from both source locations.
- For a channel set up as a single-pipeline channel, MediaLive expects the upstream system to provide one source and will therefore attempt to pull from one source location.



Setting up for a MediaConnect input

This section describes how to create a MediaConnect input. Create the input before you create the channel that ingests the input.

With a MediaConnect input, the service provider pushes content through MediaConnect to MediaLive. (From the point of view of MediaLive, the upstream system is MediaConnect. The upstream system is not the service provider.)

To perform this setup, you must work with an AWS Elemental MediaConnect user.

Topics

Step 1: Set up AWS Elemental MediaConnect

A MediaConnect user must set up AWS Elemental MediaConnect with flows to deliver source content to AWS Elemental MediaLive.

To set up flows for a standard channel

1. Provide the MediaConnect user with this information:

   - Information about the provider of the source content.
   - The AWS Region for the channel that you that will create. The AWS Elemental MediaConnect flows and the MediaLive channel (and input) must be in the same Region.

     If the flows and the MediaLive channel aren't in the same Region, then the MediaConnect operator will have to set up a distribution to move the source content to the same Region as the MediaLive input.

2. Discuss with the MediaConnect user whether you need new flows:

   - You need new flows if the source content doesn't yet have flows in MediaConnect.
   - You can reuse existing flows so long as you follow these rules:
     - Each flow doesn't exceed its maximum output bandwidth.
     - Each flow doesn't exceed its maximum number of outputs from the flow. (MediaLive automatically creates an output on each flow after you create the input in the next step, the section called “Step 2: Create input”.)

3. If you decide you need new flows, ask the MediaConnect user to create two flows.

   - They should assign flow names that are identical except for a suffix. For example, **`sports_event_A`** and **`sports_event_B`**. These suffixes will help you, the MediaLive user, to match the flows to the input pipelines in MediaLive.
   - They should set up each flow in a different Availability Zone. (If the flows are in the same Availability Zone then you, the MediaLive user, won't be able to create the MediaLive inputs.)
   - They should speak to the service provider about the following:
     - To determine how to complete the source information for each flow.
     - To make sure that the service provider delivers two sources.
     - To make sure that the two sources have identical video resolution and bitrate.
   - They should not create outputs or entitlements.

4. Obtain the following information from the MediaConnect user:

   - The ARNs for the flows. For example:

     ```
     arn:aws:mediaconnect:us-west-1:111122223333:flow:1bgf67:sports_event_A
     ```

     ```
     arn:aws:mediaconnect:us-
     west-1:111122223333:flow:9pmlk76:sports_event_B
     ```

     Note that the ARNs include the flow names as the last portion.

To set up flows for a single-pipeline channel

1. Provide the MediaConnect user with this information:

   - Information about the provider of the source content.

- The AWS Region for the channel that you will create. The AWS Elemental MediaConnect flow and the MediaLive channel (and input) must be in the same Region.

  If the flow and the MediaLive channel aren't in the same Region, then the MediaConnect operator will have to set up a distribution to move the source content to the same Region as the MediaLive input.

2. Discuss with the MediaConnect user whether you need a new flow:

   - You need a new flow if the source content doesn't yet have a flow in MediaConnect.
   - You can reuse an existing flow so long as you follow these rules:
     - The flow doesn't exceed its maximum output bandwidth.
     - The flow doesn't exceed its maximum number of outputs from the flow. (MediaLive automatically creates an output on the flow after you create the input in the next step, the section called “Step 2: Create input”.)
3. If you decide you need a new flow, ask the MediaConnect user to create one flow.

   - They should speak to the service provider to determine how to complete the source information for the flow.
   - They should not create an output or entitlement.
4. Obtain the ARN for the flow from the MediaConnect user. For example:

   ```
   arn:aws:mediaconnect:us-west-1:111122223333:flow:1bgf67:sports_event_A
   ```

   Note that the ARN includes the flow name as the last portion.

Step 2: Create a MediaConnect input

After MediaConnect is set up, you can create the MediaConnect input. A MediaLive user performs this step.

Create your input before you create the channel that ingests the input.

To create an input in MediaLive

1. Make sure that you have the information from step 1.
2. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
3. In the navigation pane, choose **Inputs**. On the **Inputs** page, choose **Create input**.

4. Complete the **Input details** section:

   - **Input** name – enter a name.
   - **Input type** – choose **MediaConnect**.

5. Complete the **MediaConnect flows** section:

   - **Channel and input class** – choose the class for this input:
     - STANDARD_INPUT
     - SINGLE_INPUT
   - **ARN for flow A** – specify the ARN for the flow that you identified as the first flow.

     If you created a second flow, then for **ARN for flow B**, specify the ARN for the second flow.

6. Complete the **Role ARN** section to choose a role for MediaLive to use with this input. For information, see the section called “IAM role and ARN”.
7. In the **Tags** section, create tags if you want to associate tags with this input. For more information, see the section called “Tagging resources”.
8. Choose **Create**.

   MediaLive creates the input and automatically creates two endpoints on that input. MediaLive always creates two endpoints, even if you specified only one flow (flow A) for the input.
9. At the same time, MediaLive automatically connects to the MediaConnect flows.

   - If you specified two flows for the input, MediaLive instructs AWS Elemental MediaConnect to create two outputs and attach them to the two flows that you created in the first stage.
   - If you specified only one flow for the input (to support a single-pipeline channel), MediaLive instructs AWS Elemental MediaConnect to create one output and to attach it to the single flow that you created in the first stage.

   If MediaConnect has two flows for the channel, it runs the flows in different Availability Zones—one zone for flow A, another zone for flow B. Similarly, MediaLive runs each pipeline in a different Availability Zone—one zone for pipeline A, another zone for pipeline B.

   MediaLive coordinates with AWS Elemental MediaConnect to ensure that MediaLive runs the channel pipelines in the same two Availability Zones as AWS Elemental MediaConnect. This setup ensures maximum resiliency if one flow fails.

IAM role and ARN

This section describes how to complete the **Role ARN** section on the **Create input** pane of the MediaLive console.

You must choose a role for MediaLive to assume when it creates any input. The role ensures that MediaLive succeeds in its request to MediaConnect to create outputs on the flows. MediaLive sends this request as soon as you choose **Create** for this input.

> **Note**
>
> This section on the MediaLive console is identical to the **IAM role** section on the **Create channel** page (also on the MediaLive console). The difference in the two usages is that on the **Create input** page, you are attaching the role to the input. On the **Create channel** page, you are attaching the role to the channel. You can use the same role (for example, the **MediaLiveAccessRole**) in both usages.

There are two general scenarios for choosing a role, depending on whether your organization has a designated administrator.

Your organization has a designated administrator

Your organization might have an administrator who manages this service. That administrator has likely set up one or more roles:

- Ask the administrator or your manager which role to use. Or if only one role is listed in **Use existing role**, choose that role.
- If the only role that is listed is **MediaLiveAccessRole**, choose that role. In addition, if the **Update** button is displayed beside this role name, choose the button. (The button does not always appear, but whenever it does appear, choose it to refresh the role.)
- If you want the selected role to appear first in the list next time, select **Remember ARN**.

Your organization has no administrator

Your organization might not have a designated service administrator. In this case, if none of your colleagues have set up a suitable role, you might have to create one yourself and then choose it.

- You can create the default role, called **MediaLiveAccessRole**. To first check if someone else has already created this role (only one person needs to create it for all users in your AWS account), look at **Create role from template**:
  - If this option is grayed out, this task has been done. In that case, choose Use existing role, and then choose **MediaLiveAccessRole** from the list.
  - If this option is not grayed out, choose **Create role from template**, and then choose **Create IAM role**. Next, choose that role from the list. If MediaLive does not let you create the role, speak to an administrator in your organization about your IAM permissions.
- If the **MediaLiveAccessRole** has already been created and the **Update** button is displayed beside it, choose the button. (The button does not always appear, but whenever it does appear, choose it to refresh the role.)
- If you want the selected role to appear first in the list next time, select **Remember ARN**.

Result of this procedure

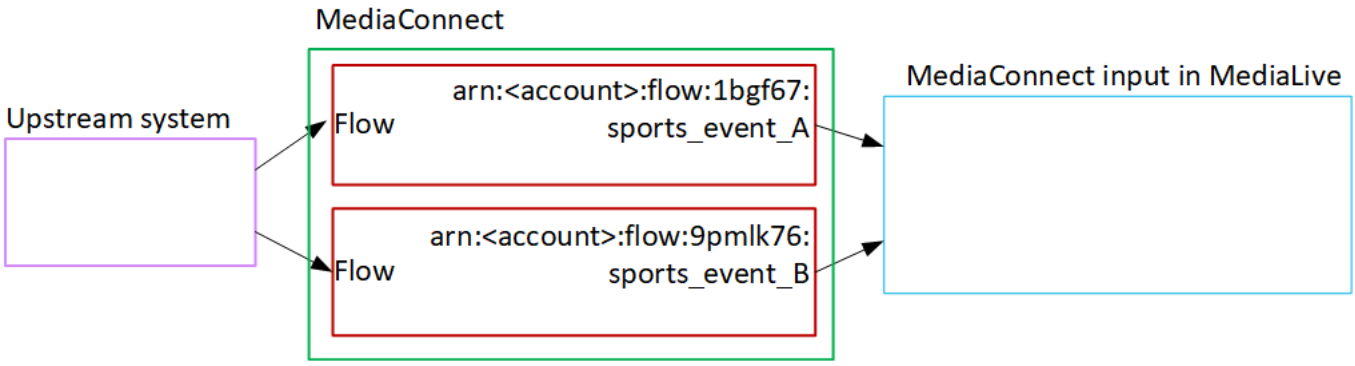
The results of this setup are illustrated in the diagram that follows. There are three main components:

- The upstream system (purple box)
- One or two MediaConnect flows (red boxes).
- One MediaConnect input in MediaLive.

Each MediaConnect flows has a source that the upstream system is pushing to. Each flow also has one output for the use of MediaLive.

The MediaConnect input in MediaLive specifies the ARNs for those outputs.

The upstream system pushes the source content to the source on the AWS Elemental MediaConnect flow or flows. The flows push the content to MediaLive. At runtime of the channel, MediaLive reacts to the content that is being pushed and ingests it.



Setting up an MP4 input

This section describes how to set up the source content on the upstream system, and how to create an MP4 input that connects the content source to AWS Elemental MediaLive. Create the input before you create the channel that ingests the input.

With an MP4 input, MediaLive connects to the upstream system when the channel starts and *pulls* the sources.

To perform this setup, you must work with an operator at the upstream system.

Topics

Step 1: Obtain information

Obtain the following information from the operator at the upstream system:

- The URLs on the upstream system for the source file or files.

  There are two URLs for a standard-class input, or one URL for a single-class input. For information about input classes and their uses, see the section called “Channel classes and input classes”.

  See the table later in this section for the URL format and for examples.

Make a note of the full URLs.

- The user name and password to access the upstream system, if the upstream system requires authenticated requests. Note that these user credentials relate to user authentication, not to the protocol. User authentication is about whether the upstream system will accept your request. The protocol is about whether the request is sent over a secure connection.

The following tables show the format of the URLs on the different types of upstream systems that MediaLive supports for MP4 input.

Upstream server is an HTTP or HTTPS server

Format of URL	*<protocol>* //:*<hostname>* /*<filename>* .mp4
Example	https://203.0.113.13/filler-videos/oceanwaves.mp4 https://198.51.100.54/filler-videos/oceanwaves.mp4

Upstream server is AWS Elemental MediaStore

Format of URL	**mediastoressl:// *<data endpoint for container>* /*<path>*/*<filename>*. mp4**
Example	Assume that the data endpoint for the container for one of the content sources is the following: **f31z.data.mediastore.us-west-2.amazonaws.com** . Assume that the file is called oceanwaves.mp4 , and it is stored in the container, in the path filler-video . The URL for one of the source files would be:

	mediastoressl://f31z.data.mediastore.us-west-2.amazonaws.com/filler-video/oceanwaves.mp4

Upstream server is Amazon S3

Upstream server	Format of URL
Format of URL	s3ssl://*<bucket>*/*<path>*/*<filename>* .mp4
Example	s3ssl://DOC-EXAMPLE-BUCKET/filler-videos/main/oceanwaves.mp4 s3ssl://DOC-EXAMPLE-BUCKET/filler-videos/redundant/oceanwaves.mp4 With MediaLive, the S3 bucket name must not use dot notation. For example, EXAMPLE-BUCKET is acceptable but EXAMPLE.BUCKET isn't.

Step 2: Create an MP4 input

After you have obtained information from the upstream system, you can create an MP4 input.

To create an MP4 pull input

1. Make sure that you have the information from step 1.
2. If this input is being used in a multiple-input channel, you should have decided whether to set it up as a static input or a dynamic input. You might need to modify the URLs you obtained from the upstream system:

   - If the input is a static input, don't modify the URLs.

- If the input is a dynamic input, set up the URL as an optional absolute portion and a required variable portion ($urlPath$). For examples, see the table after this procedure.

  We recommend that you use the format <protocol>/$urlPath$.

3. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
4. In the navigation pane, choose **Inputs**. On the **Inputs** page, choose **Create input**.
5. Complete the **Input details** section:

   - **Input** name – enter a name.
   - **Input type** – choose **MP4**.
6. In the **Input class** section, choose the class for this input:

   - STANDARD_INPUT
   - SINGLE_INPUT
7. In the **Input sources** section, enter the URLs you previously obtained:

   - If the input is a standard-class input, complete both fields, to provide two URLs.
   - If the input is a single-class input, complete the first field with the URL that you obtained and leave the second field empty.

   If the upstream system requires that you provide user credentials, you must also enter the user name and password key for accessing the location. These credentials are stored on the Systems Manager Parameter Store. For more information, see the section called "About the feature for creating password parameters".
8. In the **Tags** section, create tags if you want to associate tags with this input. For more information, see the section called "Tagging resources".
9. Choose **Create**.

   MediaLive creates the input and adds it to the list of inputs. The input specifies either one or two sources. The sources don't appear in the list, but if you choose the **Name** link, the details page shows them.

   When you start the channel, MediaLive will connect to the upstream system at this source location or locations and pull the content:

- For a standard channel, MediaLive expects the upstream system to provide two sources and will therefore attempt to pull from both source locations.
- For a single-pipeline channel, MediaLive expects the upstream system to provide one source and will therefore attempt to pull from one source location.

Formats for the URL in a dynamic input

The following table describes the different formats for the URL in a dynamic input.

Format	Description	Example	Example of the $urlPath$
<protocol>/$urlPath$	URL has only the protocol in the absolute portion	s3ssl://$urlPath$	DOC-EXAMPLE-BUCKET/my-movie.mp4
<protocol and path>/$urlPath$	URL has the protocol and path in the absolute portion	mediastoressl://f31z.data.mediastore.us-west-2.amazonaws.com/movies/$urlPath$	my-movie.mp4
$urlPath$	URL has only the variable portion	$urlPath$	s3ssl://DOC-EXAMPLE-BUCKET/my-movie.mp4

Step 3: Ensure correct setup on the MP4 upstream system

An operator at the upstream server must set up the source content on the upstream system. Make sure that the operator sets up as follows:

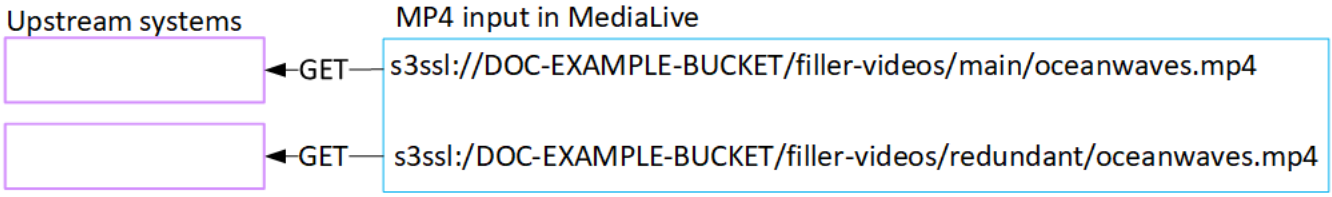
- They set up to deliver the correct number of sources:
  - If the MediaLive channel is a standard channel, the operator must set up two file sources. They must make sure that the two files are identical in terms of video resolution and bitrate.
  - If the MediaLive channel is a single-pipeline channel, the operator must set up one file source.

- They set up to make the content available at the agreed URLs. These URLs are the URLs that you obtained earlier in this section, and that you configured into the MP4 input. They correspond to the URLs shown in the diagram after this procedure.

Result of this procedure

As a result of this setup, a MediaLive input exists that specifies one or two *source* URLs. These sources are the URLs for the source content on the upstream server.

At runtime of the channel, MediaLive will connect to those two URLs (for a standard channel) or one URL (for a single-pipeline channel), and pull the source content into MediaLive.



Setting up an RTMP pull input

This section describes how to set up the source content on the upstream system, and how to create an RTMP pull input that connects the upstream system to MediaLive. Create the input before you create the channel that ingests the input.

With an RTMP Pull input, MediaLive connects to the upstream system when the channel starts and *pulls* the sources.

To perform this setup, you must work with an operator at the upstream system.

Topics

Step 1: Obtain information

Obtain the following information from your contact person at the upstream system:

- The application name and application instance for the source content. (The application instance is also known as the *stream* or *stream key*.) There are two sources for a standard-class input , or one source for a single-class input. For information about input classes and their uses, see the section called "Channel classes and input classes". For information about input classes and their uses, see the section called "Channel classes and input classes".

  The operator of the upstream system might already have rules for assigning these names. If not, you might have names that you would like to use. Make sure that you and the operator of the upstream system are clear about these names.

  In this example, the application name and instance name are identical. But they could be different:

  Application name: `live`, and instance name `curling`

  Application name: `live`, and instance name `curling`
- The public IP addresses that MediaLive will pull the source content from.

  These addresses must include port 1935. For example:

  `rtmp://203.0.113.13:1935`

  `rtmp://198.51.100.54:1935`
- The user name and password to access the upstream system, if the upstream system requires authenticated requests. Note that these user credentials relate to user authentication, not to the protocol. User authentication is about whether the upstream system will accept your request. The protocol is about whether the request is sent over a secure connection.

Create an RTMP pull input

After you have obtained information from the upstream system, you can create an HLS input.

To create an RTMP pull input

1. Make sure that you have the information from step 1.
2. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
3. In the navigation pane, choose **Inputs**. On the **Inputs** page, choose **Create input**.
4. Complete the **Input details** section:

- **Input** name – enter a name.
- **Input type** – choose **RTMP (pull)**.

5. In the **Input class** section, choose the class for this input:

   - STANDARD_INPUT
   - SINGLE_INPUT

6. In the **Input sources** section, enter the URLs you previously obtained:

   - If the input is a standard-class input, complete both fields, to provide two URLs.
   - If the input is a single-class input, complete the first field with the URL that you obtained and leave the second field empty.

   For example:

   ```
   rtmp://203.0.113.13:1935/live/curling
   ```

   If the upstream system requires that you provide user credentials, you must also enter the user name and password key for accessing the location. These credentials are stored on the Systems Manager Parameter Store. For more information, see the section called “About the feature for creating password parameters”.

7. In the **Tags** section, create tags if you want to associate tags with this input. For more information, see the section called “Tagging resources”.

8. Choose **Create**.

   MediaLive creates the input and adds it to the list of inputs. The input specifies either one or two sources. The sources don't appear in the list, but if you choose the **Name** link, the details page shows them.

   When you start the channel, MediaLive will connect to the upstream system at this source location or locations and pull the content:

   - If you will set up the channel as a standard channel, MediaLive expects the upstream system to provide two sources and will therefore attempt to pull from both source locations.
   - If you will set up the channel as a single-pipeline channel, MediaLive expects the upstream system to provide one source and will therefore attempt to pull from one source location.

Ensure correct setup on the RTMP upstream system

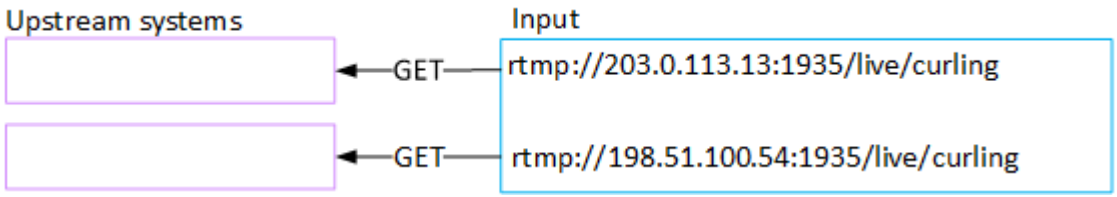
An operator at the upstream server must set up the source content on the upstream system. Make sure that the operator sets up as follows:

- They set up to deliver the correct number of sources:
  - If the MediaLive channel is a standard channel, set up two sources for the content. Make sure that the two source contents are identical in terms of video resolution and bitrate.
  - If the MediaLive channel is a single-pipeline channel, set up one source for the content.
- They set up to make the content available at the agreed URLs, and they use the agreed application names and instance names. These URLs are the URLs that you obtained earlier in this section, and that you configured into the RTMP input. They correspond to the URLs shown in the diagram after this procedure.

Result of this procedure

As a result of this setup, an RTMP pull input exists that specifies one or two *source* URLs. These sources are the URLs for the source content on the upstream system.

At runtime of the channel, the input will connect to two URLs (for a standard channel) or one URL (for a single-pipeline channel), and pull the source content identified by the application name and instance name into MediaLive.



Setting up an RTMP push input

This section describes how to set up an upstream system that uses the RTMP Push protocol to deliver source content from the public internet. It describes how to set up the source content on the upstream system, how to create an input security group, and how to create an input that connects the upstream system to MediaLive.

Create the input before you create the channel that ingests the input.

With an RTMP Push input, the upstream system *pushes* the content to MediaLive.

To perform this setup, you must work with an operator at the upstream system.

Topics

Step 1: Obtain information

Obtain the following information from your contact person at the upstream system:

- The application name and application instance for the source content. (The application instance is also known as the *stream* or *stream key*.) There are two sources for a standard-class input, or one source for a single-class input. For information about input classes and their uses, see the section called "Channel classes and input classes". For information about input classes and their uses, see the section called "Channel classes and input classes".

  The operator of the upstream system might already have rules for assigning these names. If not, you might have names that you would like to use. Make sure that you and the operator of the upstream system are clear about these names.

  In this example, the application name and instance name are identical. But they could be different:

  Application name: `live`, and instance name `curling`

  Application name: `live`, and instance name `curling`

- The public network IP addresses. These are the sets of IP addresses where the source or sources for the content will appear on the public network. You need this information to create the input security group.

  For example:

  - For one source: `203.0.113.19, 203.0.113.58, 203.0.113.25`
  - For the other source: `198.51.100.19, 198.51.100.59, 198.51.100.21`

These addresses are the addresses shown in the red boxes in the diagram after this procedure.

Step 2: Create an input security group

You must create an input security group. The security group must allow the *public network IP addresses* to push to MediaLive. Following from the earlier example, it must allow these addresses:

203.0.113.19, 203.0.113.58, 203.0.113.25, 198.51.100.19, 198.51.100.59, 198.51.100.21

For details about creating an input security group, see the section called "Creating an input security group".

Step 3: Create an RTMP push input

After you have created the input security group, you can create the RTMP push input.

To create an RTMP push input

1. Make sure that you have the information from step 1.
2. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
3. In the navigation pane, choose **Inputs**.
4. On the **Inputs** page, choose **Create input**.
5. Complete the **Input details** section:
   - **Input** name – enter a name.
   - **Input type** – choose **RTMP (push)**.
6. In the **Network mode** section, choose **Public**.
7. In the **Input security group** section, specify the group to attach to this push input. You can choose an existing group, or you can create a group. The security group must allow the public network IP addresses to push to MediaLive. Following from the example in step 1, it must allow these addresses:

   203.0.113.19, 203.0.113.58, 203.0.113.25, 198.51.100.19, 198.51.100.59, 198.51.100.21

   For more information about security groups, see the section called "Input security groups".
8. In the **Channel and input class** section, choose the class for this input:
   - STANDARD

- SINGLE-PIPELINE

For more information, see the section called "Pipeline redundancy".

9. In the **Input destinations** section, in the **Destination** section, enter the application names and application instances you previously obtained:

   - If the input is a standard-class input, complete both fields, to specify two sources.
   - If the input is a single-class input, complete the first field with the information that you obtained and leave the second field empty.

   For example:

   Application name: `live`

   Application instance: `curling`

10. In the **Tags** section, create tags if you want to associate tags with this input. For more information, see the section called "Tagging resources".

11. Choose **Create**.

    MediaLive creates the input and automatically creates two endpoints on that input. The endpoints include the application name, the application instance, and the port 1935. For example:

    ```
    198.51.100.99:1935/live/curling
    ```

    ```
    192.0.2.18:1935/live/curling
    ```

    Note that the IP addresses are addresses that MediaLive creates. They aren't the public addresses that you used in the security group. For a diagram that shows the role of all the IP addresses, see the section called "Result of this procedure" in the section about setting up an RTMP push source.

    MediaLive always creates two endpoints:

    - If you will set up the channel as a standard channel, both endpoints will be used.
    - If you will set up the channel as a single-pipeline channel, only the first endpoint will be used. MediaLive won't expect to receive content at the second endpoint.

12. Provide the upstream system with the following information:

    - If you will set up the channel as a standard channel, provide both locations. The upstream system must push the video streams to these locations.
    - If you will set up the channel as a single-pipeline channel, provide only the first location. The upstream system must push its one stream to this location.

    For example, provide these addresses:

    `198.51.100.99:1935/live/curling`

    `192.0.2.18:1935/live/curling`

Result of this procedure

As a result of this setup, an RTMP push input exists that specifies two URLs. These URLs are fixed for the lifetime of the input, regardless of changes that occur (such as modifying other information in the input, or attaching the input to a different channel).

The upstream system pushes the source content to these endpoints.

Keep in mind that with a push input, the upstream system must be pushing the video source to the input when you start the channel. The upstream system does not need to be pushing before then.

For a description of this setup that includes a diagram, see the section called "Result of this procedure" in the section about setting up an RTMP push source.

Step 4: Ensure correct setup on the upstream system

You must make sure that the upstream system pushes content to the correct locations in MediaLive.

To set up for a standard channel

Follow this procedure if the MediaLive channel is a standard channel.

1. Provide the operator with this information:

    - The two endpoints (URLs) that MediaLive generated when you created the RTMP input. These endpoints are the addresses in the blue boxes in the diagram after this procedure. The URLs include port 1935. For example:

198.51.100.99:1935/live/curling

192.0.2.18:1935/live/curling

2. Make sure that the operator sets up properly for a single-pipeline channel or a standard channel.

   If your channel is a single-pipeline channel, the operator delivers only one source, even though the input is a standard (dual-pipeline) input. The operator must do the following:

   - Deliver one source.
   - Make sure that the sources appear on the agreed IP addresses on the public network. For example:
     - The sources could appear on these addresses: 203.0.113.19, 203.0.113.58, 203.0.113.25
     - The operator can ignore the other addresses: 198.51.100.19, 198.51.100.59, 198.51.100.21

     You used these addresses when you created the input security group. If the upstream system doesn't use these addresses, MediaLive will refuse the push.
   - Push to one URL on MediaLive, and use the agreed application name and instance name. For example:

     Push to this URL: 198.51.100.99:1935/live/curling

     Ignore the other URL: 192.0.2.18:1935/live/curling

   If your channel is a standard channel, the operator must do the following:

   - Deliver two sources that are identical in terms of video resolution and bitrate.
   - Make sure that the sources appear on the agreed IP addresses on the public network. For example:
     - For one source: 203.0.113.19, 203.0.113.58, 203.0.113.25
     - For the other source: 198.51.100.19, 198.51.100.59, 198.51.100.21

     You used these addresses when you created the input security group. If the upstream system doesn't use these addresses, MediaLive will refuse the push.

- Push to the correct URLs on MediaLive, and use the agreed application name and instance name. For example, they must push to:

  ```
  198.51.100.99:1935/live/curling
  ```

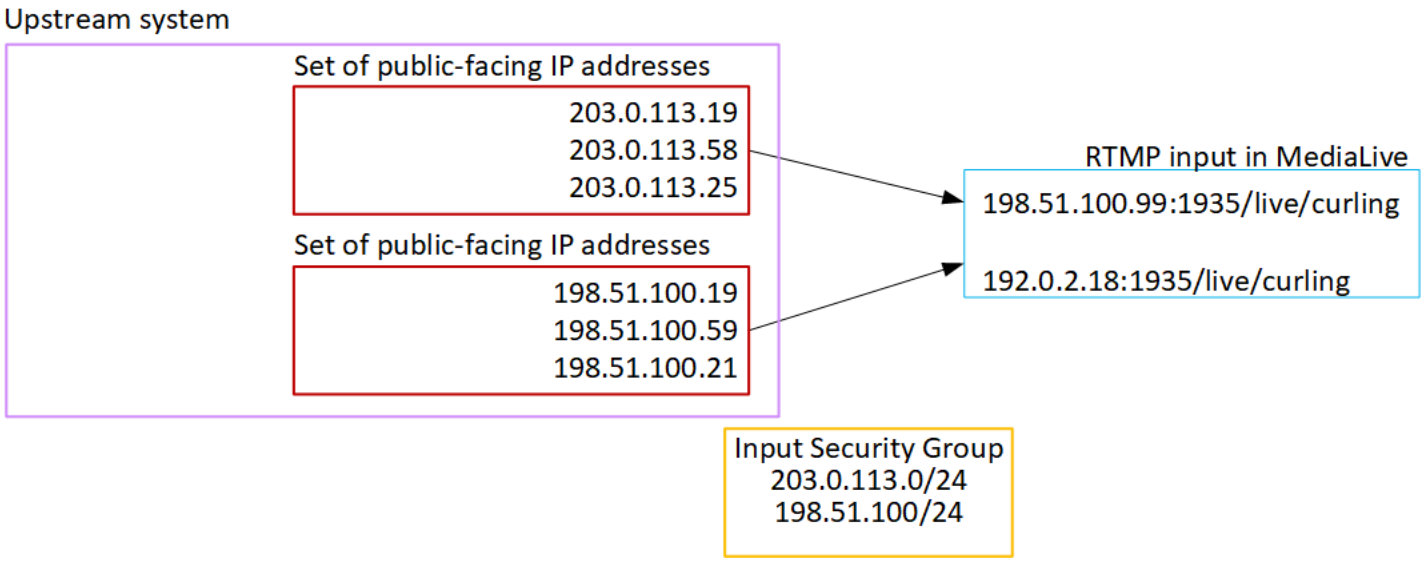
  ```
  192.0.2.18:1935/live/curling
  ```

Result of this procedure

As a result of this setup, an RTMP push input exists that specifies one or two *endpoint* URLs. These endpoints are on MediaLive.

The upstream system has been set up to push the source content to the two endpoints (for a standard channel) or to the first endpoint (for a single-pipeline channel). An input security group has been associated with the input. This input security group has a CIDR block that covers the IP addresses where the pushed source will appear on the public network, which ensures that MediaLive accepts the pushed content.

At runtime of the channel, MediaLive reacts to the content that is being pushed and ingests it.



Setting up an RTMP VPC input

This section describes how to set up content that uses the RTMP Push protocol to deliver source content from an upstream system that is in your VPC from Amazon Virtual Private Cloud (Amazon

VPC). This section describes how to set up the source content on the upstream system, and how to create an input that connects the upstream system to MediaLive.

Create the input before you create the channel that ingests the input.

With an RTMP Push input, the upstream system *pushes* the content to MediaLive.

To perform this setup, you must work with an Amazon VPC user, and with an operator at the upstream system.

Topics

Step 1: Request setup on the VPC

An Amazon VPC user must set up the VPC, and identify subnets and security groups that the upstream system and MediaLive will use.

To set up the VPC

1. Provide the Amazon VPC user with the following guidelines.

   - Guideline for the subnets – Request two subnets.

     These rules apply:

     - The two subnets must be in different Availability Zones.
     - Each subnet must have a private CIDR block (a range of IP addresses).
     - Each subnet must have at least two unused addresses in that block—one for the upstream system and one for the RTMP input.
     - Any other VPC-based sources (source B) that you create for use in the same channel as this RTMP source (source A) must be in subnets that are in the same Availability Zones as source A. The two subnets of the source B can be different from the source A, but the Availability Zones of those two subnets must be the same as the Availability Zones of source A.

- Guideline for the security group – The security group or groups for each subnet must follow these rules:
  - The combined rules of the security groups must allow inbound traffic from the IP addresses of the upstream system in that subnet.
  - The combined rules of the security groups must allow outbound traffic to port 1935.

2. After the Amazon VPC user has performed the setup, obtain the following information:

   - The ID of the VPC. For example: `vpc-3f139646`
   - The IDs of the two subnets. For example, one subnet might have this ID: `subnet-1122aabb`
   - The IDs of the security groups for the subnet or subnets. For example: `sg-51530134`

Step 2: Create an RTMP input

After the Amazon VPC user has set up on the VPC, you can create the RTMP VPC push input in MediaLive.

To create an RTMP VPC push input

1. Make sure that you have the information from step 1.
2. You should also have obtained information from the provider of the video content: the application name and application instance for the source content. (The application instance is also known as the *stream* or *stream key*.) For example:

   Application name: `live`, and instance name `curling`
3. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
4. In the navigation pane, choose **Inputs**.
5. On the **Inputs** page, choose **Create input**.
6. Complete the **Input details** section:
   - **Input** name – enter a name.
   - **Input type** – choose **RTMP (push)**.
7. In the **Network mode** section, choose **VPC**.
8. Complete the **VPC settings** section:

- Choose **Select subnets and security groups**.
- For **Subnets**, choose one of the subnets that you obtained. The dropdown list shows subnets in all VPCs, identified as follows:

  ```
  <subnet ID> <Availability Zone of subnet> <IPv4 CIDR block of subnet>
  <VPC ID> <Subnet tag called "Name", if it exists>
  ```

  For example:

  subnet-1122aabb us-west-2a 10.1.128.0/24 vpc-3f139646 Subnet for MLive push inputs

  If the list of subnets is empty, choose **Specify custom VPC**, and enter the subnet ID in the field. (You need to enter only the subnet ID, for example, **`subnet-1122aabb`**.)
- In **Subnets**, choose the second subnet. This second time, the dropdown list shows only the subnets in the same VPC as the first subnet.
- For **Security groups**, choose the security group or groups that you obtained, following the same process as for the subnets. The dropdown list shows security groups belonging to the VPC that you chose, identified as follows:

  ```
  <security group ID> <description attached to this security group> <VPC
  ID>
  ```

  For example:

  sg-51530134 Security group for MLive push inputs vpc-3f139646

9. Complete the **Role ARN** section to choose a role for MediaLive to use with this input. For more information, see the section called “IAM role and ARN”.
10. In the **Input class** section, choose the class for this input:

    - STANDARD
    - SINGLE-PIPELINE
11. In the **Input destinations** section, in the **Destination** section, enter the application names and application instances you previously set up:

    - If the input is a standard-class input, complete both fields, to specify two sources.
    - If the input is a single-class input, complete the first field with the information that you obtained and leave the second field empty.

For example:

Application name: `live`

Application instance: `curling`

12. In the **Tags** section, create tags if you want to associate tags with this input. For more information, see the section called “Tagging resources”.
13. Choose **Create**.

    MediaLive creates the input and automatically creates two endpoints on that input. These endpoints have a private IP address from the subnet range, and they specify port 1935. For example:

    `10.12.30.44:1935/live/curling`

    `10.99.39.15:1935/live/curling`
14. Provide the upstream system with these endpoints:

    - If you will set up the channel as a standard channel, provide both endpoints. The upstream system must push the content to both endpoints.
    - If you will set up the channel as a single-pipeline channel, provide only the first endpoint. The upstream system must push to this one endpoint.

    For example, provide these addresses:

    `10.12.30.44:1935/live/curling`

    `10.99.39.15:1935/live/curling`

Result of these procedures

As a result of this setup, each output of the upstream system has an IP address in one of the specified subnets in your VPC.

The RTMP input has two IP addresses. These addresses are fixed for the lifetime of the input, regardless of changes that occur (such as modifying other information in the input, or attaching the input to a different channel).

Each address is in one of those subnets. In this way, the delivery of the content from the upstream system to MediaLive takes place within the security of the VPC.

For a description of this setup that includes a diagram, see the section called "Result of this procedure" in the section about setting up an RTMP VPC source.

Keep in mind that with a push input, the upstream system must be pushing the video source to the input when you start the channel. The upstream system does not need to be pushing before then.

IAM role and ARN

This section describes how to complete the **Role ARN** section on the **Create input** pane of the MediaLive console.

You must choose a role for MediaLive to assume when it creates an RTMP Push input. To create the input, MediaLive must obtain the network interfaces for the two endpoints in the input. These endpoints are in the CIDR range of the subnets that you identified. As soon as you choose **Create** for this input, MediaLive requests these network interfaces from Amazon VPC. The role that you choose ensures that MediaLive succeeds in its request to Amazon VPC.

> **Note**
>
> This section on the MediaLive console is identical to the **IAM role** section on the **Create channel** page (also on the MediaLive console). The difference in the two usages is that on the **Create input** page, you are attaching the role to the input. On the **Create channel** page, you are attaching the role to the channel. You can use the same role (for example, the **MediaLiveAccessRole**) in both usages.

There are two general scenarios for choosing a role, depending on whether your organization has a designated administrator.

Your organization has a designated administrator

Your organization might have an administrator who manages this service. That administrator has likely set up one or more roles:

- Ask the administrator or your manager which role to use. Or if only one role is listed in **Use existing role**, choose that role.

- If the only role that is listed is **MediaLiveAccessRole**, choose that role. In addition, if the **Update** button is displayed beside this role name, choose the button. (The button does not always appear, but whenever it does appear, choose it to refresh the role.)
- If you want the selected role to appear first in the list next time, select **Remember ARN**.

Your organization has no administrator

Your organization might not have a designated service administrator. In this case, if none of your colleagues have set up a suitable role, you might have to create one yourself and then choose it.

- You can create the default role, called **MediaLiveAccessRole**. To first check if someone else has already created this role (only one person needs to create it for all users in your AWS account), look at **Create role from template**:
  - If this option is grayed out, this task has been done. In that case, choose **Use existing role**, and then choose **MediaLiveAccessRole** from the list.
  - If this option is not grayed out, choose **Create role from template**, and then choose **Create IAM role**. Next, choose that role from the list. If MediaLive does not let you create the role, speak to an administrator in your organization about your IAM permissions.
- If the **MediaLiveAccessRole** has already been created and the **Update** button is displayed beside it, choose the button. (The button does not always appear, but whenever it does appear, choose it to refresh the role.)
- If you want the selected role to appear first in the list next time, select **Remember ARN**.

Step 3: Ensure correct setup on the upstream system

You must make sure that the upstream system sets up correctly with your VPC and pushes content to the correct locations in MediaLive.

To set up for a standard channel

Follow this procedure if the MediaLive channel is a standard channel.

1. Provide the operator with this information:
   - The IDs of the VPC, two subnets, and the security groups that the Amazon VPC user gave you.

- The two endpoints (URLs) that MediaLive generated when you created the RTMP input. These endpoints are the addresses in the blue boxes in the diagram after this procedure. The URL has a private IP address and includes port 1935. For example:

  ```
  10.12.30.131:1935/live/curling
  ```

  ```
  10.99.39.40:1935/live/curling
  ```

2. Make sure that the operator sets up properly for a standard channel. They must do the following:

   - Set up two separate upstream systems. They can't set up one upstream system with two output interfaces because you, the MediaLive user, will lose the redundancy that you want to achieve with a standard channel (with two independent pipelines).
   - Set up two output interfaces—one output interface in one of the subnets, and set up the other upstream system with one output interface in the other subnet. These interfaces are the addresses in the purple boxes in the diagram after this procedure.
   - Make sure that the two content sources are identical in terms of video resolution and bitrate.
   - Push to the correct URLs on MediaLive, and use the agreed application name and instance name. For example, they must push to:

     ```
     10.12.30.131:1935/live/curling
     ```

     ```
     10.99.39.40:1935/live/curling
     ```

To set up for a single-pipeline channel

Follow this procedure if the MediaLive channel is a single-pipeline channel.

1. Provide the operator with this information:

   - The IDs of the VPC, one subnet, and the security groups that the Amazon VPC user gave you.
   - Only the first of the two endpoints (URLs) that MediaLive generated when you created the RTMP input. These endpoints are the addresses in the blue boxes in the diagram after this procedure. The URL has a private IP address and includes port 1935. For example:

     ```
     10.12.30.131:1935/live/curling
     ```

2. Make sure that the operator sets up properly for a single-pipeline channel. They must do the following:

- Set up one upstream system.
- Set up one output interface. The interface is the address in one of the purple boxes in the diagram after this procedure.
- Push to the correct URL on MediaLive. For example, they must push to:

  ```
  10.12.30.131:1935/live/curling
  ```

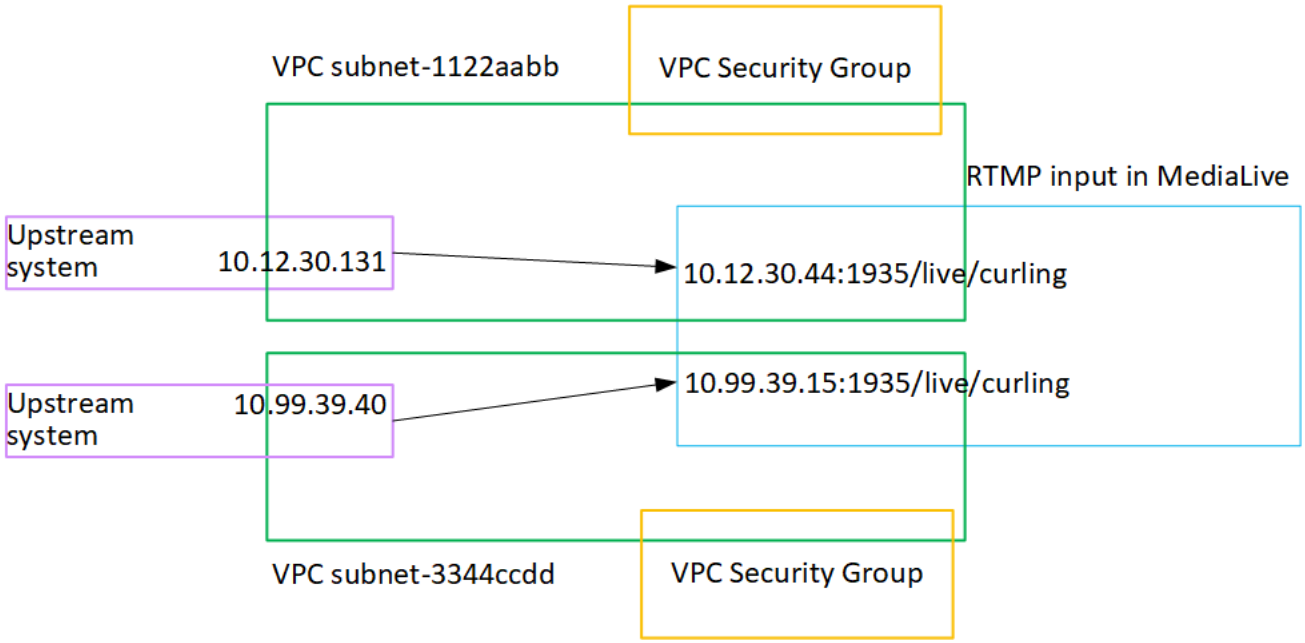
Result of this procedure

As a result of this setup, an RTMP input exists that specifies one or two *endpoint* URLs. These endpoints are elastic network interfaces on your VPC. MediaLive has permission to use these network interfaces for its inputs. MediaLive has permission (through the IAM trusted entity role) to automatically manage the network interfaces for its inputs. The upstream system has permission, through the Amazon VPC security group, to push content to these endpoints.

The upstream system or systems have been set up to push the source content to the two endpoints (if you are setting up for a standard channel) or to one endpoint (if you are setting up for a single-pipeline channel). At least one VPC security group has been associated with each subnet. The CIDR block in each security group covers the two URLs that the upstream system pushes from, which ensures that MediaLive accepts the pushed content.

Each output of the upstream system has an IP address in one of the specified subnets in your VPC. The RTMP input has two IP addresses, and each address is in one of those subnets. In this way, the delivery of the source content from the upstream system to MediaLive takes place within the privacy of the VPC.

At runtime of the channel, MediaLive reacts to the content that is being pushed and ingests it.



Setting up an RTP push input

This section describes how to set up an upstream system that uses the RTP Push protocol to deliver source content from an upstream system that is in your VPC from Amazon VPC. It describes how to set up the source content on the upstream system, and how to create an input that connects the upstream system to MediaLive.

With an RTP push source, the upstream system *pushes* the content to MediaLive.

To perform this setup, you must work with an operator at the upstream system.

Topics

Step 1: Obtain information `

Obtain the following information from your contact person at the upstream system:

- The public network IP addresses. You need two sets of IP addresses because an RTP input is always a standard-class input, even if your channel is a single-pipeline channel. For information about input classes, see the section called "Channel classes and input classes".

  These are the sets of IP addresses where the source or sources for the content will appear on the public network. You need this information to create the input security group.

  For example:

  - For one source: `203.0.113.19, 203.0.113.58, 203.0.113.25`
  - For the other source: `198.51.100.19, 198.51.100.59, 198.51.100.21`

Step 2: Create an input security group

You must create an input security group. The security group must allow the *public network IP addresses* to push to MediaLive. Following from the earlier example, it must allow these addresses:

203.0.113.19, 203.0.113.58, 203.0.113.25, 198.51.100.19, 198.51.100.59, 198.51.100.21

For details about creating an input security group, see the section called "Creating an input security group".

Step 3: Create an RTP input

After you have created the input security group, you can create the RTP push input.

To create an RTP push input

1. Make sure that you have the information from step 1.
2. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
3. In the navigation pane, choose **Inputs**. On the **Inputs** page, choose **Create input**.
4. Complete the **Input details** section:
   - **Input** name – enter a name.
   - **Input type** – choose **RTP**.
5. In the **Network mode** section, choose **Public**.
6. In the **Input security group** section, specify a group to attach to this push input. You can choose an existing group, or you can create a group. For more information about security groups, see the section called "Input security groups". The security group must allow the public

network IP addresses to push to MediaLive. Following from the example in step 1, it must allow these addresses:

203.0.113.19, 203.0.113.58, 203.0.113.25, 198.51.100.19, 198.51.100.59, 198.51.100.21

For more information about security groups, see the section called “Input security groups”.

7. In the **Tags** section, create tags if you want to associate tags with this input. For more information, see the section called “Tagging resources”.
8. Choose **Create**.

   MediaLive creates the input and automatically creates two endpoints on that input. These endpoints include the port 5000. For example:

   ```
   198.51.100.99:5000
   ```

   ```
   192.0.2.18:5000
   ```

   Note that the IP addresses are addresses that MediaLive creates. They aren't the public addresses that you used in the security group. For a diagram that shows the role of all the IP addresses, see the section called “Result of this procedure” in the section about setting up an RTP push source.

   MediaLive always creates two endpoints:

   - If you will set up the channel as a standard channel, both endpoints will be used.
   - If you will set up the channel as a single-pipeline channel, only the first endpoint will be used. MediaLive won't expect to receive content at the second endpoint.
9. Provide the upstream system with the following information:

   - If you will set up the channel as a standard channel, provide both locations. The upstream system must push the video streams to these locations.
   - If you will set up the channel as a single-pipeline channel, provide only the first location. The upstream system must push its one stream to this location.

   For example, provide these addresses:

   ```
   198.51.100.99:5000
   ```

   ```
   192.0.2.18:5000
   ```

Result of this procedure

As a result of this setup, an RTP push input exists that specifies two URLs. These URLs are fixed for the lifetime of the input, regardless of changes that occur (such as modifying other information in the input, or attaching the input to a different channel).

The upstream system pushes the source content to these endpoints.

Keep in mind that with a push input, the upstream system must be pushing the video source to the input when you start the channel. The upstream system does not need to be pushing before then.

For a description of this setup that includes a diagram, see the section called "Result of this procedure" in the section about setting up an RTP source.

Step 4: Ensure correct setup on the upstream system

You must make sure that the upstream system pushes content to the correct locations in MediaLive.

To set up for a standard channel

Follow this procedure if the MediaLive channel is a standard channel.

1. Provide the operator with this information:

   - The two endpoints (URLs) that MediaLive generated when you created the RTP input. These endpoints are the addresses in the blue boxes in the diagram after this procedure. The URLs include port 5000. For example:

     `198.51.100.99:5000`

     `192.0.2.18:5000`
2. Make sure that the operator sets up properly for a standard channel. They must:

   - Deliver two sources that are identical in terms of video resolution and bitrate.
   - Make sure that the sources appear on the agreed IP addresses on the public network. For example:
     - For one source: `203.0.113.19, 203.0.113.58, 203.0.113.25`
     - For the other source: `198.51.100.19, 198.51.100.59, 198.51.100.21`

You used these addresses when you created the input security group. If the upstream system doesn't use these addresses, MediaLive will refuse the push.

- Push to the correct URLs on MediaLive. For example, they must push to:

  ```
  198.51.100.99:5000
  ```

  ```
  192.0.2.18:5000
  ```

- Send over RTP, not UDP. The UDP protocol is not supported for an input into MediaLive.

To set up for a single-pipeline channel

Follow this procedure if the MediaLive channel is a single-pipeline channel.

1. Provide the operator with this information:

   - Only the first of the two endpoints (URLs) that MediaLive generated when you created the RTP input. This endpoint is one of the addresses in the blue boxes in the diagram after this procedure. The URL includes port 5000. For example:

     ```
     198.51.100.99:5000
     ```

2. Make sure that the operator sets up properly for a single-pipeline channel. They must:

   - Make sure that the source appears on the agreed IP addresses on the public network. For example:

     ```
     203.0.113.19, 203.0.113.58, 203.0.113.25
     ```

     You used these addresses when you created the input security group. If the upstream system doesn't use these addresses, MediaLive will refuse the push.

   - Push to the correct URL on MediaLive. For example, they must push to:

     ```
     198.51.100.99:5000
     ```

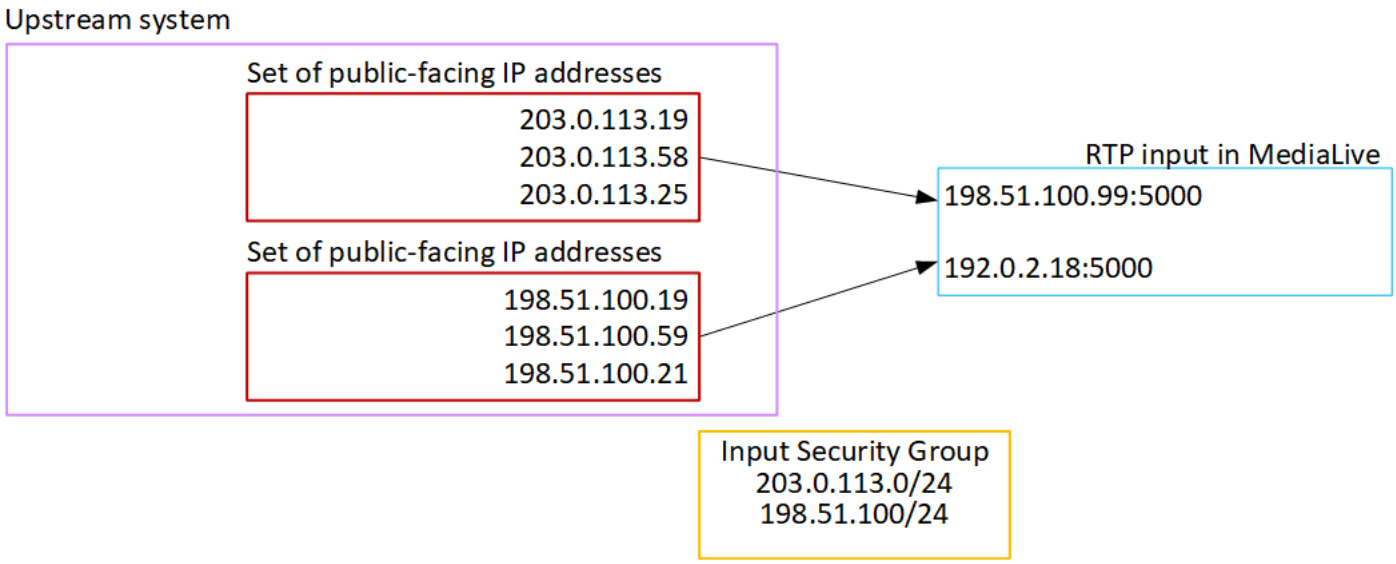
   - Send over RTP, not UDP. The UDP protocol is not supported for an input into MediaLive.

Result of this procedure

As a result of this setup, an RTP input exists that specifies one or two *endpoint* URLs. These endpoints are on MediaLive.

The upstream system has been set up to push the source content to the two endpoints (for a standard channel) or to the first endpoint (for a single-pipeline channel). An input security group has been associated with the input. This input security group has a CIDR block that covers the two URLs that the upstream system pushes, which ensures that MediaLive accepts the pushed content.

At runtime of the channel, MediaLive reacts to the content that is being pushed and ingests it.



Setting up an RTP VPC input

This section describes how to set up an upstream system that uses the RTP Push protocol to deliver source content from an upstream system that is in your Amazon Virtual Private Cloud (Amazon VPC). It describes how to set up the source content on the upstream system, and how to create an input that connects the upstream system to MediaLive.

With an RTP VPC source, the upstream system *pushes* the content to MediaLive.

To perform this setup, you must work with an Amazon VPC user, and with an operator at the upstream system.

Topics

Step 1: Request setup on the VPC

An Amazon VPC user must set up the VPC, and identify subnets and security groups that the upstream system and MediaLive will use.

To set up the VPC

1. Provide the Amazon VPC user with the following guidelines.

   - Guideline for the subnets – Request two subnets.

     These rules apply:

     - You need two subnets because an RTP input is always a standard-class input, even if your channel is a single-pipeline channel. For information about input classes, see the section called “Channel classes and input classes”.
     - The two subnets must be in different Availability Zones.
     - Each subnet must have a private CIDR block (a range of IP addresses).
     - Each subnet must have at least two unused addresses in that block—one for the upstream system and one for the RTP input.
     - Any other VPC-based sources (source B) that you create for use in the same channel as this RTP source (source A) must be in subnets that are in the same Availability Zones as source A. The two subnets of the source B can be different from the source A, but the Availability Zones of those two subnets must be the same as the Availability Zones of source A.

   - Guideline for the security group – The security group or groups for each subnet must follow these rules:

     - The combined rules of the security groups must allow inbound traffic from the IP addresses of the upstream system in that subnet.
     - The combined rules of the security groups must allow outbound traffic to port 5000.

2. After the Amazon VPC user has performed the setup, obtain the following information:

   - The ID of the VPC. For example: `vpc-3f139646`
   - The IDs of the two subnets. For example, one subnet might have this ID: `subnet-1122aabb`
   - The IDs of the security groups for the subnets. For example: `sg-51530134`

Step 2: Create an input for RTP push from Amazon VPC

After the Amazon VPC user has set up on the VPC, you can create the RTP VPC push input in MediaLive.

To create an RTP VPC push input from Amazon VPC

1. Make sure that you have the information from step 1.
   - The ID of the VPC.
   - The IDs of the two subnets.
   - The IDs of the security groups for the subnet or subnets.
2. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
3. In the navigation pane, choose **Inputs**.
4. On the **Inputs** page, choose **Create input**.
5. Complete the **Input details** section:
   - **Input** name – enter a name.
   - **Input type** – choose **RTP**.
6. In the **Network mode** section, choose **VPC**.
7. Complete the **VPC settings** section:
   - Choose **Select subnets and security groups**.
   - For **Subnets**, choose one of the subnets that you obtained. The dropdown list shows subnets in all VPCs, identified as follows:

     ```
     <subnet ID> <Availability Zone of subnet> <IPv4 CIDR block of subnet>
     <VPC ID> <Subnet tag called "Name", if it exists>
     ```

     For example:

     subnet-1122aabb us-west-2a 10.1.128.0/24 vpc-3f139646 Subnet for MLive push inputs

     If the list of subnets is empty, choose **Specify custom VPC**, and enter the subnet ID in the field. (You need to enter only the subnet ID, for example, **subnet-1122aabb**.)
   - In **Subnets**, choose the second subnet. This second time, the dropdown list shows only the subnets in the same VPC as the first subnet.

- For **Security groups**, choose the security group or groups that you obtained, following the same process as for the subnets. The dropdown list shows security groups belonging to the VPC that you chose, identified as follows:

  `<security group ID> <description attached to this security group> <VPC ID>`

  For example:

  sg-51530134 Security group for MLive push inputs vpc-3f139646

8. Complete the **Role ARN** section to choose a role for MediaLive to use with this input. For more information, see the section called “IAM role and ARN”.
9. In the **Tags** section, create tags if you want to associate tags with this input. For more information, see the section called “Tagging resources”.
10. Choose **Create**.

    MediaLive creates the input and automatically creates two endpoints on that input. These endpoints have a private IP address from the subnet range, and they specify port 5000. For example:

    `rtp://10.12.30.44:5000`

    `rtp://10.99.39.15:5000`.
11. Provide the upstream system with these endpoints:

    - If you will set up the channel as a standard channel, provide both endpoints. The upstream system must push the content to both endpoints.
    - If you will set up the channel as a single-pipeline channel, provide only the first endpoint. The upstream system must push to this one endpoint.

Result of these procedures

As a result of this setup, each output of the upstream system has an IP address in one of the specified subnets in your VPC.

The RTP input has two IP addresses. These addresses are fixed for the lifetime of the input, regardless of changes that occur (such as modifying other information in the input, or attaching the input to a different channel).

Each address is in one of those subnets. In this way, the delivery of the content from the upstream system to MediaLive takes place within the security of the VPC.

For a description of this setup that includes a diagram, see the section called "Result of this procedure" in the section about setting up an RTP VPC source.

Keep in mind that with a push input, the upstream system must be pushing the video source to the input when you start the channel. The upstream system does not need to be pushing before then.

IAM role and ARN

This section describes how to complete the **Role ARN** section on the **Create input** pane of the MediaLive console.

You must choose a role for MediaLive to assume when it creates an RTP Push input. To create the input, MediaLive must obtain the network interfaces for the two endpoints in the input. These endpoints are in the CIDR range of the subnets that you identified. As soon as you choose **Create** for this input, MediaLive requests these network interfaces from Amazon VPC. The role that you choose ensures that MediaLive succeeds in its request to Amazon VPC.

> **Note**
>
> This section on the MediaLive console is identical to the **IAM role** section on the **Create channel** page (also on the MediaLive console). The difference in the two usages is that on the **Create input** page, you are attaching the role to the input. On the **Create channel** page, you are attaching the role to the channel. You can use the same role (for example, the **MediaLiveAccessRole**) in both usages.

There are two general scenarios for choosing a role, depending on whether your organization has a designated administrator.

Your organization has a designated administrator

Your organization might have an administrator who manages this service. That administrator has likely set up one or more roles:

- Ask the administrator or your manager which role to use. Or if only one role is listed in **Use existing role**, choose that role.

- If the only role that is listed is **MediaLiveAccessRole**, choose that role. In addition, if the **Update** button is displayed beside this role name, choose the button. (The button does not always appear, but whenever it does appear, choose it to refresh the role.)
- If you want the selected role to appear first in the list next time, select **Remember ARN**.

Your organization has no administrator

Your organization might not have a designated service administrator. In this case, if none of your colleagues have set up a suitable role, you might have to create one yourself and then choose it.

- You can create the default role, called **MediaLiveAccessRole**. To first check if someone else has already created this role (only one person needs to create it for all users in your AWS account), look at **Create role from template**:
  - If this option is grayed out, this task has been done. In that case, choose **Use existing role**, and then choose **MediaLiveAccessRole** from the list.
  - If this option is not grayed out, choose **Create role from template**, and then choose **Create IAM role**. Next, choose that role from the list. If MediaLive does not let you create the role, speak to an administrator in your organization about your IAM permissions.
- If the **MediaLiveAccessRole** has already been created and the **Update** button is displayed beside it, choose the button. (The button does not always appear, but whenever it does appear, choose it to refresh the role.)
- If you want the selected role to appear first in the list next time, select **Remember ARN**.

Step 3: Ensure correct setup on the upstream system

You must make sure that the upstream system sets up correctly with your VPC and pushes content to the correct locations in MediaLive.

To set up for a standard channel

Follow this procedure if the MediaLive channel is a standard channel.

1. Provide the operator with this information:
   - The IDs of the VPC, two subnets, and the security groups that the Amazon VPC user gave you.

- The two endpoints (URLs) that MediaLive generated when you created the RTP input. These endpoints are the addresses in the blue boxes in the diagram after this procedure. The URLs have private IP addresses and include port 5000. For example:

  ```
  10.12.30.44:5000
  ```

  ```
  10.99.39.15:5000
  ```

2. Make sure that the operator sets up properly for a standard channel. They must:

   - Set up two output interfaces—one output interface in one of the subnets, and set up the other upstream system with one output interface in the other subnet. These interfaces are the addresses in the purple boxes in the diagram after this procedure.
   - Deliver two sources that are identical in terms of video resolution and bitrate.
   - Push to the correct URLs on MediaLive. For example, they must push to:

     ```
     10.12.30.131:5000
     ```

     ```
     10.99.39.40:5000
     ```
   - Send over RTP, not UDP. The UDP protocol is not supported for an input into MediaLive.

To set up for a single-pipeline channel

Follow this procedure if the MediaLive channel is a single-pipeline channel.

1. Provide the operator with this information:

   - The IDs of the VPC, one subnet, and the security groups that the Amazon VPC user gave you.
   - Only the first of the two endpoints (URLs) that MediaLive generated when you created the RTP input. These endpoints are the addresses in the blue boxes in the diagram after this procedure. The URL has a private IP address and includes port 5000. For example:

     ```
     10.12.30.44:5000
     ```

     ```
     10.99.39.15:5000
     ```

2. Make sure that the operator sets up properly for a standard channel. They must:

   - Set up one output interface. The interface is the address in one of the purple boxes in the diagram after this procedure.

- Push to the correct URL on MediaLive. For example, they must push to:

  ```
  10.12.30.131:5000
  ```

  ```
  10.99.39.40:5000
  ```

- Send over RTP, not UDP. The UDP protocol is not supported for an input into MediaLive.

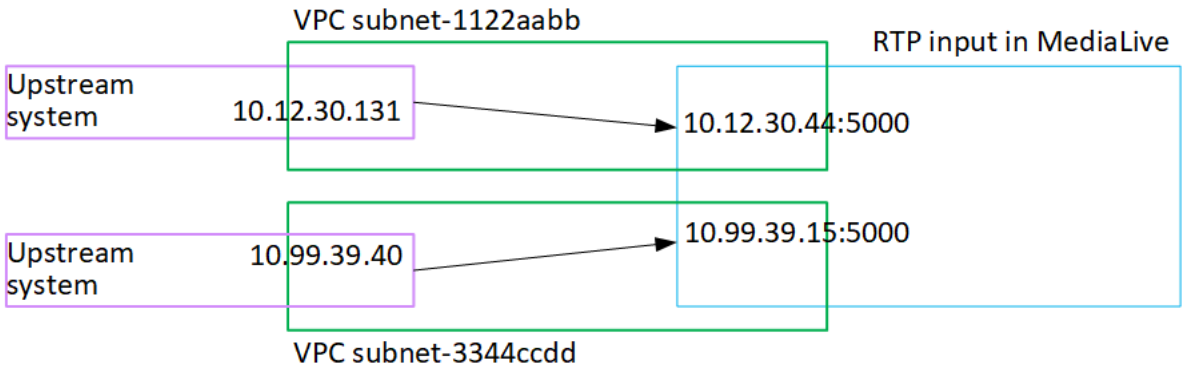
Result of this procedure

As a result of this setup, an RTP input exists that specifies one or two *endpoint* URLs. These endpoints are elastic network interfaces (ENIs) on your VPC. MediaLive has permission to use these ENIs for its inputs. MediaLive has permission (through the IAM trusted entity role) to automatically manage the ENIs for its inputs. The upstream system has permission, through the Amazon VPC security group, to push content to these endpoints.

The upstream system or systems have been set up to push the source content to the two endpoints (if you are setting up for a standard channel) or to one endpoint (if you are setting up for a single-pipeline channel). At least one VPC security group has been associated with each subnet. The CIDR block in each security group covers the two URLs that the upstream system pushes from, which ensures that MediaLive accepts the pushed content.

Each output of the upstream system has an IP address in one of the specified subnets in your VPC. The RTP input has two IP addresses, and each address is in one of those subnets. In this way, the delivery of the source content from the upstream system to MediaLive takes place within the privacy of the VPC.

At runtime of the channel, MediaLive reacts to the content that is being pushed and ingests it.



Creating a transport stream (TS) file input

Create your input before you create the channel that ingests the input.

To create a TS file input

1. You should have already arranged with the video content provider to set up the upstream system for your content. Make sure that the operator of the upstream system gives you the following information:

   - The full URLs of the locations where MediaLive will pull the TS files from. For example:

     ```
     s3ssl://DOC-EXAMPLE-BUCKET/filler-videos/main/oceanwaves.ts
     ```

     ```
     s3ssl://DOC-EXAMPLE-BUCKET/filler-videos/redundant/oceanwaves.m2ts
     ```

2. If this input is being used in a multiple-input channel, you should have decided whether to set it up as a static input or a dynamic input. You might need to modify the URLs you obtained from the upstream system:

   - If the input is a static input, don't modify the URLs.
   - If the input is a dynamic input, set up the URL as an optional absolute portion and a required variable portion ($urlPath$). For examples, see the table after this procedure.

     We recommend that you use the format <protocol>/$urlPath$.

3. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
4. In the navigation pane, choose **Inputs**. On the **Inputs** page, choose **Create input**.
5. Complete the **Input details** section:

   - **Input** name – enter a name.
   - **Input type** – choose **TS**.

6. In the **Input class** section, choose the class for this input:

   - STANDARD_INPUT
   - SINGLE_INPUT

7. In the **Input sources** section, enter the URLs you previously obtained:

   - If the input is a standard-class input, complete both fields, to provide two URLs.

- If the input is a single-class input, complete the first field with the URL that you obtained and leave the second field empty.

If the upstream system requires that you provide user credentials, you must also enter the user name and password key for accessing the location. These credentials are stored on the Systems Manager Parameter Store. For more information, see the section called "About the feature for creating password parameters".

8. In the **Tags** section, create tags if you want to associate tags with this input. For more information, see the section called "Tagging resources".
9. Choose **Create**.

MediaLive creates the input and adds it to the list of inputs. The input specifies either one or two sources. The sources don't appear in the list, but if you choose the **Name** link, the details page shows them.

When you start the channel, MediaLive will connect to the upstream system at this source location or locations and pull the content:

- For a standard channel, MediaLive expects the upstream system to provide two sources and will therefore attempt to pull from both source locations.
- For a single-pipeline channel, MediaLive expects the upstream system to provide one source and will therefore attempt to pull from one source location.

Formats for the URL in a dynamic input

The following table describes the different formats for the URL in a dynamic input.

Format	Description	Example	Example of the $urlPath$
<protocol>/$urlPath$	URL has only the protocol in the absolute portion	s3ssl://$urlPath$	DOC-EXAMPLE-BUCKET/my-movie.ts
<protocol and path>/$urlPath$	URL has the protocol and path in the absolute portion	mediastoressl://f31z.data.mediastore.us-west-2.amazona	my-movie.ts

Format	Description	Example	Example of the $urlPath$
		ws.com/movies/$urlPath$	
$urlPath$	URL has only the variable portion	$urlPath$	s3ssl://DOC-EXAMPLE-BUCKET/my-movie.ts

Next steps

After you have created all the inputs that you need for a channel, you are ready to start creating the channel. See *Setup: Creating a channel* .

Creating a channel from scratch

A channel ingests and transcodes (decodes and encodes) source content from the inputs that are attached to that channel, and packages the new content into outputs. You create and configure the channel with the details that instruct the channel how to perform this processing.

There are three ways to create a channel:

- **Create from scratch.** See the topics in this chapter.
- **Use a built-in or custom template.** See the section called “Creating a channel from a template”.
- **Clone an existing channel.** See the section called “Creating a channel by cloning”.

Important

The information in this chapter is presented in sections, typically one section for each part of the **Channel** page on the MediaLive console. You can complete the sections in any order. You can complete part of a section and then come back to it later.
MediaLive validates all the information once, after you save the channel.

Topics

Getting ready

We recommend that before you start creating the channel, you plan the workflow and plan the channel. In both these planning procedures, you obtain information that you need to create the

channel. In addition, you must create the inputs that you need. You won't be able to create the channel unless you have created these inputs.

Here is the information that you need, listed in the order in which your will use it when you create the channel:

- You need the following information in the section called "Channel and input details":
  - Whether you will implement any resiliency features of MediaLive, and particularly whether you will create a standard channel or a single-pipeline channel. You made these decisions in step 3 of *Setup: Preparing upstream and downstream*.
- You need the following information in the section called "Inputs part 1: Attach inputs":
  - The names of the input or inputs to use in this channel. You created the input or inputs in *Setup: Creating inputs*.
- You need the following information to create the input selectors, as part of the procedure in the section called "Inputs part 2: Configure inputs":
  - The assets to extract from each input. You identified these assets in the section called "Step 2: Map outputs to sources", as part of planning the channel.
- You need the following information in the section called "Outputs":
  - The output groups to create. You should have identified these output groups in step 1 of *Setup: Preparing upstream and downstream*.
  - The outputs to create. You should have designed the outputs and encodes (video, audio,and captions) when you planned the channel.
  - Information about the destinations for the outputs of each output group. You obtained this information in step 7 of *Setup: Preparing upstream and downstream*.
- You need the following information in the three steps that start with the section called "Set up video":
  - Details about the output encodes (video, audio, and captions) to create in each output group. You made these decisions in *Setup: Planning the channel*.

> **Note**
>
> For information about additional steps for setting up a channel for use in a multiplex program, see the section called "Setting up a multiplex".

Complete channel and input details

The **Channel and input details** section of the **Create channel** page lets you do the following:

- Select the IAM role that AWS Elemental MediaLive will use to access the channel when the channel is running (started).
- Optionally select a template to use.
- Select the channel class.
- Complete input specification information.
- Set up tagging.

To provide channel and input details

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. Before creating a channel, make sure that you have created the inputs that you will attach to the channel.
3. On the MediaLive home page, choose **Create channel**, and in the navigation pane, choose **Channels**.

   If you've created a channel before, you won't see the home page. In that case, in the MediaLive navigation pane, choose **Channels**, and then choose **Create channel**.
4. On the **Create channel** page, choose **Channel and input details**.
5. Complete the sections:

   - In **General info**, for **Channel name**, type a name for your channel.
   - In **General info**, complete **IAM role**. See the section called “IAM role and ARN”.
   - For information about the **Channel template** section, see the section called “Creating a channel from a template”.
   - In **Channel class**, choose the class. See the section called “Channel class”.
   - In **Input specifications**, complete the fields to match your input. See the section called “Input specifications settings”.
   - In the **Tags** section, create tags if you want to associate tags with this channel. For more information, see the section called “Tagging resources”.
6. When ready, go to the next section.

IAM role and ARN

This section describes how to complete the **IAM role** section in the **General info** section of the **Channel and input details** pane.

You must choose a role for MediaLive to assume when it works with this channel. If you don't choose a role, you can't create the channel. There are two general scenarios, depending on whether your organization has a designated administrator.

> **Note**
>
> This section on the MediaLive console is identical to the **IAM role** section on the **Create input** page for a MediaConnect push input (also on the MediaLive console). The difference in the two usages is that on the **Create channel** page, you attach the role to the channel. On the **Create input** page, you attach the role to the MediaConnect input. You can use the same role (for example, the **MediaLiveAccessRole**) in both usages.

There are two general scenarios for choosing a role, depending on whether your organization has a designated administrator.

Your organization has a designated administrator

Your organization might have an administrator who manages this service. That administrator has likely set up one or more roles:

- Ask the administrator or your manager which role to use. Or if only one rule is listed in **Use existing role**, choose that role.
- If the only rule that is listed is **MediaLiveAccessRole**, choose that role. In addition, if the **Update** button is displayed beside this role name, choose the button. (The button does not always appear, but whenever it does appear, choose it to refresh the role.)
- If you want the selected ARN to appear first in the list next time, select **Remember ARN**.

Your organization has no administrator

Your organization might not have a designated service administrator. In this case, if none of your colleagues have set up a suitable role, you might have to create one yourself and then choose it.

- You can create the default role, called **MediaLiveAccessRole**. To first check if someone else has already created this role (only one person needs to create it for all users in your AWS account), look at **Create role from template**:
  - If this option is grayed out, this task has been done. In that case, choose **Use existing role**, and then choose **MediaLiveAccessRole** from the list.
  - If this option is not grayed out, choose **Create role from template**, and then choose **Create IAM role**. Next, choose that role from the list. If MediaLive does not let you create the role, speak to an administrator about your IAM permissions.
- If the **MediaLiveAccessRole** has already been created and the **Update** button is displayed beside it, choose the button. (The button does not always appear, but whenever it does appear, choose it to refresh the role.)
- If you want the selected ARN to appear first in the list next time, select **Remember ARN**.

Channel class

When you planned the workflow, you decided whether to set up the channel as a standard channel (with two pipelines) or a single-pipeline channel. You must now specify the class in the channel configuration.

For **Channel class**, choose `STANDARD` or `SINGLE_PIPELINE`.

Standard class

With this class, the channel contains two pipelines. The input for the channel has two entry points. The upstream system sends identical source streams to these two entry points, to provide content to two pipelines within the channel. MediaLive performs identical processing on both pipelines. For each output that you configure (for example, for both HLS output and RTMP output), the two pipelines deliver identical content to two destinations on the downstream system.

Single pipeline class

With this class, the channel contains one pipeline. For each output that you configure, the channel delivers content to one destination on the downstream system.

Input specifications settings

The **Input Specifications** settings include three fields that characterize the video in the input that you intend to use with this channel. The fields are the following:

- Input codec
- Input resolution
- Maximum input bitrate

You should have obtained information about these video characteristics when you assessed the upstream system for each input source.

To complete the settings

1. In your list of planned inputs, look at all the inputs except for any Elemental Link inputs. Find the following codec, resolution, and bitrate:

   - Find the most resource-intensive codec among all the inputs. The codecs, from least to most intensive, are MPEG2, then AVC, then HEVC. Make a note of the codec. The input it appears in isn't relevant.
   - Find the highest resolution tier among all the inputs. The tiers, from lowest to highest, are SD, HD, UHD. Make a note of the tier. The input it appears in isn't relevant.
   - Find the highest bitrate among all the inputs. Make a note of the bitrate. The input it appears in isn't relevant.

2. For each field, choose an option that meets or exceeds the value you identified for that field.

   Follow these tips:

   - If your channel contains only one input and it is from an AWS Elemental Link device, leave the input specification fields with their defaults. Make sure that you configure the **Input resolution** in the device. That field has the same purpose as these input specifications fields.
   - If you aren't sure about the processing requirements of your inputs, choose a higher option. For example, if you aren't sure of the bitrate and you are trying to choose between 10 Mbps and 20 Mbps, then choose 20 Mbps, to be on the safe side. Or if you aren't sure if your inputs use AVC (H.264) or HEVC (H.265), then choose HEVC.

How MediaLive uses this information

MediaLive uses these values for billing and resource allocation purposes.

- For billing, MediaLive uses these fields to calculate the charges that you will incur on the input side. You pay for the option that you specify. For example, if you specify HD but the inputs are all actually SD, you will still be charged for HD.
- For resource allocation, MediaLive uses these fields to allocate processing resources when you run this channel. If you don't choose the correct option, MediaLive might not allocate sufficient processing resources. Insufficient processing resources might mean that your channel output starts to degrade when the channel is running.

MediaLive doesn't use these values to determine what is actually in the video for decoding purposes. At ingest time, it still inspects the video to detect the source codec, resolution, and bitrate.

Billing and resource allocation for Elemental Link input

MediaLive doesn't use the input specification as for the charge per minute for Elemental Link input usage. Instead, MediaLive uses the resolution that you specify in the device configuration.

Attach inputs to the channel

Before you started to create the channel, you should have created all of inputs for all the content sources for the channel.

You must now attach the inputs to the channel. You can attach multiple inputs to the channel. For detailed information about setting up a channel with more than one input, see the section called “Input switching”. There are specific rules about the number and type (push versus pull, for example) of inputs that you can attach to one channel.

The procedure to attach inputs

To attach one input

1. On the **Create channel** page, for **Input attachments**, choose **Add**.
2. On the **Attach input** section, for **Input**, choose an existing input. After you choose the input, information about the input appears.

   To review this information, see the following sections:

   - the section called “Channel input—CDI VPC push input ”

the section called “Channel input—Elemental Link push input”

- the section called “Channel input—HLS pull input”
- the section called “Channel input—MediaConnect push input”
- the section called “Channel input—MP4 pull input”
- the section called “Channel input—RTMP push input”
- the section called “Channel input—RTMP pull input”
- the section called “Channel input—RTP push input”

3. For **Attachment name**, enter a name for the attachment. The default name is the name of the input itself.
4. Choose **Confirm**. The **Input attachment** section closes, and the **General input settings** section appears.
5. For information about completing the fields in the **General input settings** section, go to the next step.

Channel input—CDI VPC push input

To verify that the input is set up correctly, look at the **Input destinations** section. It shows the two locations on MediaLive that the upstream system will push the source to when the channel is running. These locations were automatically generated when you created the input:

- If the channel is set up as a standard channel, two locations are generated.
- If the channel is set up as a single-pipeline channel, one location is generated.

For example:

10.99.39.23:5000

192.0.2.54:5000

Channel input—Elemental Link push input

To view the status of the AWS Elemental Link hardware device for this input, look at the **Details**. If the device is currently pushing content to MediaLive, the **Device** thumbnail shows the content. The device generates the thumbnails by capturing a video frame approximately every 5 seconds.

Channel input—HLS pull input

To verify that the input is set up correctly, look at the **Input sources** section. It shows the locations of the source video. You specified these locations when you created the input:

- If the channel is set up as a standard channel, you specified two locations.
- If the channel is set up as a single-pipeline channel, you specified one.

For example, for an HTTPS pull:

https://203.0.113.13/sports/curling.m3u8 and

https://203.0.113.54/sports/curling.m3u8

Or, for a pull from an AWS Elemental MediaStore container:

mediastoressl://eri39n.data.mediastore.us-west-2.amazonaws.com/sports/canada/curling.m3u8

and

mediastoressl://21lu05.data.mediastore.us-west-2.amazonaws.com/sports/canada/curling.m3u8

Or, for a pull from an Amazon S3 bucket:

s3ssl://DOC-EXAMPLE-BUCKET/filler-videos/main/oceanwaves.mp4 and

s3ssl://DOC-EXAMPLE-BUCKET/filler-videos/redundant/oceanwaves.mp4

Channel input—MediaConnect push input

To verify that the input is set up correctly, look at the **MediaConnect flows** section. It shows the ARNs of the AWS Elemental MediaConnect flows that are the source for this input. These ARNs were automatically generated when you created the input:

- If the channel is set up as a standard channel, two ARNs are generated.
- If the channel is set up as a single-pipeline channel, one ARN is generated.

For example:

arn:aws:mediaconnect:us-west-1:111122223333:flow:1bgf67:sports-event-A and

arn:aws:mediaconnect:us-west-1:111122223333:flow:9pmlk76:sports-event-B

Channel input—MP4 pull input

To verify that the input is set up correctly, look at the **Input destinations** section. It shows the locations of the source video. You specified these locations when you created the input:

- If the channel is set up as a standard channel, you specified two locations.
- If the channel is set up as a single-pipeline channel, you specified one.

The format of the location depends on the type of upstream system:

- For an upstream system that uses HTTP or HTTPS, the location is an HTTP or HTTPS URL. For example:

  https://203.0.113.31/filler-videos/oceanwaves.mp4

  https://203.0.113.52/filler-videos/oceanwaves.mp4
- For a file that is stored on Amazon S3, the location is the bucket name and object for the file. For example:

  s3ssl://DOC-EXAMPLE-BUCKET/filler-videos/main/oceanwaves.mp4

  s3ssl://DOC-EXAMPLE-BUCKET/filler-videos/redundant/oceanwaves.mp4

Channel input—RTMP pull input

To verify that the input is set up correctly, look at the **Input destinations** section. It shows the locations of the source video. You specified these locations when you created the input:

- If the channel is set up as a standard channel, you specified two locations.
- If the channel is set up as a single-pipeline channel, you specified one.

For example:

rtmp://203.0.113.13:1935/live/curling/

rtmp://198.51.100.54:1935/live/curling/

Channel input—RTMP push input

Follow these guidelines to verify that the input is set up correctly.

To verify the setup of the input

1. Look at the **Input destinations** section. It shows the two locations on MediaLive that the upstream system will push the source to when the channel is running. These locations were automatically generated when you created the input:

   - If the channel is set up as a standard channel, two locations are generated.
   - If the channel is set up as a single-pipeline channel, one location is generated.

   Each location consists of an address portion that was automatically generated, appended by a folder that you specified when you created the input.

   For example, for an RTMP Public push input:

   rtmp://198.51.100.99:1935/live/curling

   rtmp://192.0.2.18:1935/live/curling

   For example, for an RTMP VPC push input:

   rtmp://10.12.30.44:1935/live/curling

   rtmp://10.99.39.15:1935/live/curling
2. Look again at the **Input destinations** section.

   - If the section has an **Input security group** with a number beside it, then the input is an RTMP Public input that has a MediaLive security group. The input is correctly set up and you can continue.
   - If the section has an **Input security group** without a number beside it, then the input is an RTMP Public input that is missing a MediaLive input security group. This input isn't correctly set up. Typically, this situation occurs if, for example, you have input A attached to input security group B and then you delete B. Input A is no longer useable. You must recreate the

input and attach an input security group to it before you can associate it with a channel that you are creating.

- If the section doesn't have an **Input security group**, then the input is an RTMP VPC push input. The input is correctly set up and you can continue.

Channel input—RTP push input

Follow these guidelines to verify that the input is set up correctly.

To verify the setup of the input

1. Look at the **Input destinations** section. It shows the two locations on MediaLive that the upstream system will push the source to when the channel is running. These locations were automatically generated when you created the input:

   - If the channel is set up as a standard channel, two locations are generated.
   - If the channel is set up as a single-pipeline channel, one location is generated.

   For example, for an RTP Public input:

   rtp://198.51.100.99:5000

   rtp://192.0.2.18:5000

   For example, for an RTP VPC input:

   rtp://10.12.30.44:5000

   rtp://10.99.39.15:5000
2. Look again at the **Input destinations** section.

   - If the section has an **Input security group** with a number beside it, then the input is an RTP Public input that has a MediaLive security group. The input is correctly set up and you can continue.
   - If the section has an **Input security group** without a number beside it, then the input is an RTP Public input that is missing a MediaLive input security group. This input isn't correctly set up. Typically, this situation occurs if, for example, you have input A attached to input security group B and then you delete B. Input A is no longer useable. You must recreate the

input and attach an input security group to it before you can associate it with a channel that you are creating.

- If the section doesn't have an **Input security group**, then the input is an RTP VPC input. The input is correctly set up and you can continue.

Complete the settings for each input

As soon as you attach the input on the **Attach input** sections, the **Input attachment** section closes and the **General input settings** section appears for that input. You must complete these fields to configure the input:

- Configure the input connection.

- Identify the video, audio, and captions to extract from the input.

To configure the input

1. Complete the fields as required. See the topics links below. For details about a field, choose the **Info** link next to the field on the MediaLive console:

   - For most fields, the default values are sufficient.
   - However, if you want to include audio and captions in the outputs, you must complete the **Audio selectors** and **Caption selectors** sections; the defaults do not specify enough information.
2. Complete the following field in the **General settings** section in the navigation pane:

   - **Global configuration** - **Input loss behavior**. These fields configure how the channel behaves when it stops receiving content from any input. For more details, see the section called "Input loss handling". These fields apply to all inputs, therefore you only need to set them once for the entire channel.
3. If you are setting up the channel with multiple inputs, add more inputs to the channel. For guidelines about implementing input switching, see the section called "Input switching".
4. When ready, go to the next step.

Topics

- Input settings—Network input settings
- Input settings—Other settings
- Input settings—Video selector
- Input settings—Audio selectors
- Input settings—Caption selectors

Input settings—Network input settings

The fields in the Network input settings section apply only to HLS inputs. Complete this section on the MediaLive console only if the input is HLS. MediaLive ignores these fields for other types of inputs.

Input settings—Other settings

The fields that are not within the **Network input settings** section apply to all inputs.

Input settings—Video selector

This section lets you identify the video to extract from the input, and lets you enable the optional color space feature.

To identify the video and color space

1. In **Video selector**, choose **Video selector**. More fields appear.
2. Complete **Selector settings** as specified in the table after this procedure.

   Keep in mind that there is no button to add more video selectors because you can extract only one video asset from the input.
3. (Optional) Complete **Color space** and **Color space usage**. These fields let you configure the optional color space feature. For more information, see the section called "Video – complex color space conversion".

Determining whether you need to create a video selector

When you planned the channel, you should have identified the video that you need to extract from this input.

You must now determine if you need to create a *video selector*, to identify the specific asset to extract from the input. Some input types require selectors, some input types don't require them.

The following table specifies whether you need to create a video selector.

Input type	Method of extracting video
CDI	Don't complete **Selector settings**. MediaLive extracts the first video that it encounters in the source content.
Elemental Link	The input contains only one video asset. MediaLive extracts that video. There is no need to complete **Selector settings.**
HLS	Don't complete **Selector settings**. These extraction methods don't apply to HLS inputs. By default, MediaLive extracts the video asset with the highest bandwidth. You can complete the **Bandwidth** field (in **Input settings** – **Network input settings**). MediaLive extracts the highest bandwidth video that is below this limit.
MediaConnect	If the input contains an MPTS, choose **Selector settings** , and enter the program or PID to extract. If you don't specify the program or PID, MediaLive extracts the first video it finds. If the input contains an SPTS, MediaLive extracts that video. There is no need to complete **Selector settings**.
MP4	The input contains only one video asset. MediaLive extracts that video. There is no need to complete **Selector settings.**

Input type	Method of extracting video
RTMP	The input contains only one video asset. MediaLive extracts that video. There is no need to complete **Selector settings**.
RTP	If the input contains an MPTS, choose **Selector settings** and enter the program or PID to extract. If you don't specify the program or PID, MediaLive extracts the first video it finds. If the input contains an SPTS, MediaLive extracts that video. There is no need to complete **Selector settings**.

Input settings—Audio selectors

If you want to extract audio from the input, this section is required. You create one or more audio selectors to identify the audio asset to extract. Typically, you identify different languages from the input, but you could also extract different audio codecs (such as AAC and Dolby).

You can create a maximum of 20 audio selectors in one channel.

To identify the audio to extract

1. Decide if you need to create any audio selectors. When you planned the channel, you should have identified the audio assets that you need to extract from this input.

   The following table specifies whether you need to create an audio selector in order to extract that audio. In the table, find your input type, and read the guidance.

Input type	Need to create selector?	Number of selectors to create
CDI	Yes.	One for each audio asset you want to extract. You don't have to extract every audio asset from the input.

Input type	Need to create selector?	Number of selectors to create
Elemental Link	Yes, if you want to select specific audio tracks from the input. If no audio selector is created, MediaLive will encode the input audio according to your channel's configured audio outputs.	One for each audio asset you want to extract. You don't have to extract every audio asset from the input.
HLS	Yes, if the input contains more than one audio asset, or if you aren't sure how many audios it contains.	One for each audio asset you want to extract. You don't have to extract every audio asset from the input.
MediaConnect	Yes, if the input contains more than one audio asset, or if you aren't sure how many audio assets it contains.	One for each audio asset you want to extract. You don't have to extract every audio asset from the input.
MP4	Yes, if the input contains more than one audio asset, or if you aren't sure how many audio assets it contains.	One for each audio asset you want to extract. You don't have to extract every audio asset from the input.
RTMP	No, because the input contains only one audio asset. MediaLive extracts that audio.	None

Input type	Need to create selector?	Number of selectors to create
RTP	Yes, if the input contains more than one audio asset, or if you aren't sure how many audio assets it contains.	One for each audio asset you want to extract. You don't have to extract every audio asset from the input.

If the input contains more than one audio asset and you don't create a selector, MediaLive selects the first audio it encounters.

2. Choose **Add audio selector** once for each audio that you want to extract from the input.
3. In each audio selector, in **Audio selector name**, enter a name that describes the audio that you are extracting.
4. In each audio selector, complete **Selector settings** as specified in the following table.

Input type	How to complete Selector settings
CDI	Choose **Audio track selection**, then choose **Add tracks** to add a selector for each track you want to extract. In each **Track** field, enter the track number.
Elemental Link	Choose **Audio track selection**, then choose **Add tracks** to add a selector for each track you want to extract. In each **Track** field, enter the track number.
HLS	Select in one of these ways: • Choose **Audio pid selection** and enter the PID for the audio asset. • Or choose **Audio language selection** and enter the three-letter ISO code for the language. Complete **Language selection**

Input type	How to complete Selector settings
	policy. For details about a field on the MediaLive console, choose the **Info** link next to the field. We recommend you select by PID. If you select by language, MediaLive selects the first instance of that language that it encounters. That might not be the language version you want.
MediaConnect	Select in one of these ways: • Choose **Audio pid selection** and enter the PID for the audio asset. • Or choose **Audio language selection** and enter the three-letter ISO code for the language. Complete **Language selection policy**. For details about a field on the MediaLive console, choose the **Info** link next to the field. We recommend you select by PID. If you select by language, MediaLive selects the first instance of that language that it encounters. That might not be the language version you want.
MP4	Choose **Audio track selection**, then choose **Add tracks** to add a selector for each track you want to extract. In each **Track** field, enter the track number.
RTMP	Leave this field blank.

Input type	How to complete Selector settings
RTP	Choose **Audio pid selection** and enter the PID for the audio asset in **PID**. Or choose **Audio language selection** and enter the three-letter ISO code for the language. We recommend you select by PID. If you select by language, MediaLive selects the first instance of that language that it encounters. That might not be the language version you want.

Input settings—Caption selectors

If you want to extract captions from the input or to specify an external file as the source of the captions, this section is required. You create one or more captions selectors to identify the captions to extract. Typically, you identify different languages in each selector, but you could also identify different captions formats.

For each captions item that you want to extract or include, choose the **Add captions** selector. For detailed information about setting up input for captions, see the section called “Captions”, specifically the section called “Step 1: Set up inputs”.

Complete general settings

The **General settings** section lets you configure global settings and global features:

- Global *settings* set behavior that applies to all inputs or all outputs in the channel. You can't configure the behavior differently for different inputs or outputs.
- Global *features* set up features that are optional but that apply globally to all outputs if they are enabled.

To complete the general settings

1. On the **Create channel** page, choose **General settings**.
2. In the **General channel settings** section, set the global settings and optional features as needed. For information about each setting or feature, see the topics at the end of this procedure.
3. When you have finished working with these fields, go to the next step.

Avail blanking

Optional feature. You can set this to blank out the output video during ad avails. For more information, see the section called “SCTE-35 message processing”.

Avail configuration

Optional feature. You can modify the way that MediaLive handles SCTE-35 ad avail messages, or you can keep the default behavior. For information about the default behavior and how to modify that behavior, see the section called “SCTE-35 message processing”.

Blackout slate

Optional feature. You can black out the output video as specified by program metadata, if that metadata is present in the input. For more information, see the section called “SCTE-35 message processing”.

Feature activations

Optional features. You can enable the input prepare feature for input switching. For more information, see the section called “Input prepare”.

Global configuration

Global configuration settings. In this section, complete the first three fields as appropriate. For details about each field, choose the **Info** link next to the field.

Global configuration – input loss behavior

Global configuration settings. The **Input Loss Behavior** fields control how MediaLive handles input loss. You can customize the handling. For more information, see the section called "Input loss handling".

Motion graphics configuration

Optional feature. You can enable the motion graphics overlay feature. For more information, see the section called "Motion graphics overlay".

Nielsen configuration

Optional feature. You can configure a MediaLive channel to convert Nielsen watermarks to ID3 metadata. For more information see the section called "Nielsen watermarks to ID3".

Timecode configuration

Global configuration settings. This section lets you specify the timecode for the output. For more information about configuring the timecode, see the section called "Timecodes".

Logging

Optional feature. You can enable logging of activity on this individual channel. For detailed information about this feature, see the section called "CloudWatch Logs".

To enable logging, choose a log level other than **DISABLED**. The levels are listed from least to most verbose.

To disable logging, choose **DISABLED**.

Configure outputs

The **Outputs** section lets you create output groups in the channel. The **Outputs** section provides access to the following content:

- The output group, which contains settings that configure the output packages and output containers in the channel.
- One or more outputs. Outputs are containers for video, audio, and captions output encodes.

- The individual output encodes.

For information about setting up the contents of an output group, see the following sections:

- *Setup: Creating outputs*
- the section called “Set up video”
- the section called “Set up audio”
- the section called “Set up captions”

After you have set up all the output groups, you will be ready to save the channel.

Save the channel

You can save the channel only after you have configured and created everything that you need.

To save (create) the channel, choose **Create channel** in the navigation pane.

As soon as you choose **Create channel**, MediaLive validates the configuration of the channel and displays messages for any errors. You can't save a draft of the channel, and you can't save a channel that contains error messages.

To find your newly created channel, in the navigation pane, choose **Channel**. (The navigation pane might be collapsed. To open it, choose the menu icon in the upper-left corner of the console).

The **Channel** pane appears and shows the newly created channel in the list of channels. The state changes to **Creating**, and then to **Ready**.

Next step

For your next step, we recommend that you read the chapter about the channel schedule. It is likely that there are features of the schedule that you want to use. For more information, see *Setup: Creating a schedule*.

In particularly, if you have attached more than one input to the channel, you must implement input switching in order to move from one input to another. Input switching uses schedule actions. See the section called “Input switching”.

After you have set up the schedule, you can start the channel.

Setup: Creating output groups and outputs in a channel

This section describes how to create output groups and outputs. You should have already determined the output groups that you want to create. And you should have identified the outputs and encodes to include in each output group.

You create output groups and outputs when you create or edit a channel. When you create a channel, you must create at least one output group. After you have created the channel, you can edit it to add more output groups.

On the console, you create output groups on the **Outputs** section of the **Channel** page. You can't create the output groups and outputs separately from the channel that they belong to.

Topics

Creating an archive output group

When you planned the workflow for your channel, you might have determined that you want to include an Archive output group. An Archive group always sends output to an S3 bucket.

Topics

Procedure to create an archive output group

Follow these steps to create an archive output group and output.

To create an archive output group and its output

1. On the **Create channel** page, under **Output groups**, choose **Add**.
2. In the **Add output group** section, choose **Archive**, and then choose **Confirm**. More sections appear:
   - **Archive group destination** – This section contains fields for the output destination.
   - **Archive settings** – This section contains fields for the output destination.
   - **Archive outputs** – This section shows the output that is added by default. An archive output can contain only one output, so don't click **Add output**.
3. In **Archive outputs**, choose the **Settings** link to view the sections for the individual output:
   - **Output settings** – This section contains fields for the output destination and the output container.
   - **Stream settings** – This section contains fields for the output streams (the video, audio, and captions).
4. (Optional) Enter names for the output group and the output:
   - In **Archive settings**, for **Name**, enter a name for the output group. This name is internal to MediaLive; it doesn't appear in the output. For example, **`Sports Game 10122017 ABR`** or **`tvchannel59`**.
   - In **Archive outputs**, for **Name**, enter a name for the output. This name is internal to MediaLive; it doesn't appear in the output.
5. To complete the other fields, see the topics listed after this procedure.
6. After you have finished setting up this output group and its single output, you can create another output group (of any type), if your plan requires it. Otherwise, go to the section called "Save channel".

Fields for the output destination

The following fields configure the location and names of the archive output files (the destination).

- **Output group** – **Archive group destination** section

- **Output group – Archive settings – CDN settings**
- **Output group – Additional settings – Rollover interval**
- **Archive outputs – Name modifier**
- **Archive outputs – Extension**

You must design the destination path or paths for the output. You must then enter the different portions of the path into the appropriate fields on the console.

Design the path for the output destination

As part of the planning for this output group, you discussed your requirements with the Amazon S3 user. You should already have the following information:

- The bucket names portion of the path for the output
- Or the full path for the output.

To design the path

If you haven't yet designed the destination path, design them now. If you've already designed the paths, go to the section called “Complete the fields on the console”.

- Design the destination path or paths, following this syntax:

  ```
  protocol bucket folders baseFilename nameModifier counter extension
  ```

  For example, for a standard channel:

  ```
  s3ssl://DOC-EXAMPLE-BUCKET/channel59/delivery/
  curling-20171012T033162.000000.m2ts
  ```

  ```
  s3ssl://DOC-EXAMPLE-BUCKET1/channel59/delivery/
  curling-20171012T033162.000000.m2ts
  ```

The following table maps each portion in the example to the portion in the syntax.

Portion of the URL	Example	Comment
protocol	s3ssl://	The protocol is always **`s3ssl://`** because the destination for an Archive output is always an S3 bucket.
bucket portion of the path	DOC-EXAMPLE-BUCKET	When you planned the workflow for the channel, you should have made sure that the S3 bucket or buckets exist. With MediaLive, the S3 bucket name must not use dot notation. For example, **`mycompany-videos`** is acceptable but **`mycompany.videos`** isn't.
folders portion of the path	channel59/delivery/	The folders can be present or not, and can be as long as you want. The folders must always end with a slash.
baseFilename	curling	Don't terminate the file name with a slash.
nameModifier	-20171012T033162	The modifier is optional for an archive output.
delimiter before the counter	.	MediaLive automatically inserts this delimiter.
counter	000000	MediaLive automatically generates this counter. Initially, this is a six-digit

Portion of the URL	Example	Comment
		number starting at 000000, and increasing by 1. So 000000, 000001, 000002 and so on. After 999999, the next number is 1000000 (seven digits), then 1000001, 1000002, and so on. Then from 9999999 to 10000000 (eight digits), and so on.
dot before the extension	.	MediaLive automatically inserts this dot.
extension	m2ts	Always **`m2ts`**.

Complete the fields on the console

To specify the location for the output

1. Enter the different portions of the destination in the appropriate fields.

Portion of the destination URL	Field	Example
protocol, bucket, folders, baseFilename	The two **URL** fields in the **Archive group destinations** section. The data before the first slash is the bucket name. The data after the last slash is the baseFilename. The data in between is the folders. Specify two destinations when the channel is set up as a standard channel, or one destination when it is set up as a single-pipeline channel.	`s3ssl://DOC-EXAMPLE-BUCKET/channel59/delivery/curling`

Portion of the destination URL	Field	Example
nameModifier	The **Name modifier** field in the **Archive outputs** section. If you choose to include a modifier, you can enter a string such as **`-high`**, to indicate a high-resolution output. Or you can enter a variable ID (such as `$dt$`) to ensure that the modifier is different for each file segment. For a list of variable data identifiers, see the section called "Variable data: Supported identifiers".	`$dft$`
extension	The **Extension** field in the **Archive outputs**. Always leave the default, **`m2ts`**.	`mt2s`

2. Leave the **Credentials** section blank in both the **Archive group destinations** sections. MediaLive has permission to write to the S3 bucket via the trusted entity. Someone in your organization should have already set up these permissions. For more information, see the section called "Reference: summary of trusted entity access".
3. Complete the **CDN settings** field only if MediaLive must set a canned ACL whenever it sends this output to the Amazon S3 bucket.

   Use of a canned ACL typically only applies if your organization is not the owner of the Amazon S3 bucket. You should have discussed the use of a canned ACL with the bucket owner when you discussed the destination for the output.
4. Complete the **Rollover interval** field in the **Archive settings** section.

For example, **300** divides the output into separate files, each with a 300 second (5 minutes) long duration.

Each time the rollover expires, MediaLive closes the current file on Amazon S3 and starts a new file using the `baseFilename`, the `nameModifier`, and a sequential counter.

The current file is visible on Amazon S3 only after it has closed.

For more information, see the examples.

Examples of destination fields for an archive output group

These examples show how to set up the fields that relate to file locations. They don't show how to set up other fields such as fields in the individual outputs.

Example 1

You want to create an archive of the streaming output from TV channel 59. You want to store the output in the S3 bucket named **`DOC-EXAMPLE-BUCKET`**, and you want to break up the stream into 5-minute chunks.

Field	Value
Rollover interval field in **Archive settings** section	**`300`**
URL in **Archive group destination A** section	**`s3ssl://DOC-EXAMPLE-BUCKET/channel59/delivery/curling`**
URL in **Archive group destination B** section	**`s3ssl://DOC-EXAMPLE-BUCKET/channel59/backup/curling`** Using *delivery* and *backup* as folder names is only an example.
Name modifier in **Archive outputs** section	**`-$dt$`**

Field	Value
	For information about identifiers for variable data (such as dt), see the section called "Variable data: Supported identifiers".
Extension in **Archive outputs** section	Leave blank to use the default (**.m2ts**).

Result: the output will be broken into files of 5 minutes (300 seconds) each. Each file will have a file name of **curling**, the time that the channel started and a counter (000000, 000001, and so on), and the file name extension. For example:

- The first file will be **curling-20171012T033162-000001.m2ts**.
- The second file will be **curling-20171012T033162-000002.m2ts**.

Each file will be stored in both **s3ssl://DOC-EXAMPLE-BUCKET/channel59/delivery** and **s3ssl://DOC-EXAMPLE-BUCKET/channel59/backup**.

A given file is not visible in Amazon S3 while it is being written. As soon as the rollover happens (or if the user stops the channel), MediaLive closes the current file. At that point, the file becomes visible.

Example 2

You want to create an archive of highlights from the curling game that are also being streamed (in a separate HLS output group). You want to create three outputs: one that has audio languages for Europe, one for audio languages for Asia, and one for audio languages for Africa. You want to store the outputs in the S3 buckets named **DOC-EXAMPLE-BUCKET1** and **DOC-EXAMPLE-BUCKET1**. You want to break up the stream into 5 minute chunks.

Field	Value
Rollover interval field in **Archive settings** section	**300**
URL in **Archive group destination A** section	**s3ssl://DOC-EXAMPLE-BUCKET1/sports-delivery/highlights/curling/10312017**

Field	Value
	In this example, the **10312017** folder is set to match today's date.
URL in **Archive group destination B** section	**s3ssl://DOC-EXAMPLE-BUCKET2/sports-delivery/highlights/curling/10312017** In this example, the paths have different bucket names.
Name modifier in **Archive outputs** section	Choose **Add output** twice: two more **Output** lines are added to this section, for a total of three lines. In each line, enter a modifier: **-audiogroup1** , **-audiogroup2** , and **-audiogroup3** .
Extension in **Archive outputs** section	Leave blank to use the default (**.m2ts**).

Result: three separate categories of files are created for each output. Each file has a file name of **10312017**, plus the modifier, the sequential counter, and the file name extension. For example:

- `10312017-audiogroup1-000000.m2ts`, `10312017-audiogroup2-000000.m2ts`, and `10312017-audiogroup3-000000.m2ts`.
- `10312017-audiogroup1-000001.m2ts`, `10312017-audiogroup2-000001.m2ts`, and `10312017-audiogroup3-000001.m2ts`.

Each file will be stored in both `s3ssl://DOC-EXAMPLE-BUCKET1/sports-delivery/highlights/curling` and `s3ssl://DOC-EXAMPLE-BUCKET2/sports-delivery/highlights/curling`.

A given file is not visible in Amazon S3 while it is being written. As soon as the rollover happens (or if the user stops the channel), MediaLive closes the current file. At that point, the file becomes visible.

Fields for the output container

The following fields relate to the packaging and delivery of the archive transport stream:

- In **Output settings** – **Container Settings** section
- In **Output settings** – **PID settings** section

For all these fields, optionally change any values. For details about a field, choose the **Info** link next to the field in the MediaLive console.

Fields for the video, audio, and captions streams (encodes)

The following fields relate to the encoding of the video, audio, and captions streams (encodes) in the output.

- **Stream settings** section

For information about creating encodes, see the following sections:

- the section called “Set up video”
- the section called “Set up audio”
- the section called “Set up captions”

Creating a CMAF Ingest output group

When you planned the workflow for your channel, you might have determined that you want to include a CMAF Ingest output group.

1. On the **Create channel** or **Edit channel** page, in **Output groups**, choose **Add**.
2. In the **Add output group** section, choose **CMAF Ingest**, and then choose **Confirm**. More sections appear:
   - **CMAF Ingest destination** – This section contains fields for the destination of the outputs. You should have obtained the URLs to enter when you planned the destinations for the HLS output group. The URL looks like this:

     ```
     https://mz82o4-1.ingest.hnycui.mediapackagev2.us-west-2.amazonaws.com/
     in/v1/curling-channel-group/1/curling-channel/
     ```

Leave the **Credentials** section empty. You don't need to enter credentials to authenticate with MediaPackage.

- **CMAF Ingest settings** – This section contains fields for configuring how segments are delivered and for configuring how various features behave. See later in this section.
- **CMAF Ingest outputs** – This section shows the single output that is added by default. You can add more outputs, and you can add video, audio, and captions encodes in each output. See later in this section.

3. After you have finished setting up this output group and its outputs, you can create another output group (of any type), if your plan requires it. Otherwise, go to the section called "Save channel".

Topics

Fields in CMAF Ingest settings section

Field	Description
Name	A name for the output group. This name is internal to MediaLive. It doesn't appear in the output. For example, **`Sports Curling`**.
SCTE35 Type	To pass through SCTE 35 messages in the output group, choose **SCTE_35_WITHOUT_SEGMENTATION**. The WITHOUT_SEGMENTATION wording indicates that each inserted SCTE 35 message will result in a new IDR in the video, but it won't result in a new segment. This handling is standard for CMAF Ingest For more information about setting up for SCTE 35, see the section called "SCTE-35 message processing".

Field	Description
Segment Length, Segment Length Units	Enter the preferred duration of segments (in milliseconds or seconds). The segments will end on the next keyframe after the specified duration, so the actual segment duration might be longer. If the units are seconds, the duration might be a fraction of the seconds.
Send Delay Msec	Number of milliseconds to delay the output from pipeline 1, when the channel starts or unpauses. (This field applies only to standard channels. The value is ignored in a single-pipeline channel.) Some packagers always ingest the first pipeline that they receive. You can therefore set a value here to ensure that pipeline 0 always arrives at the packager first.
Nielsen ID3 Behavior	For information about this feature, see the section called "Nielsen watermarks to ID3".

Fields for the video, audio, and captions streams (encodes)

1. In **CMAF Ingest outputs**, choose **Add output** to add the appropriate number of outputs to the list of outputs.
2. Choose the first **Settings** link to view the first output. Each output has two sections: **Output settings** and **Stream settings**.
3. Complete **Output settings**:
   - **Output name**: Change the randomly generated name to a meaningful name. This name is internal to MediaLive; it doesn't appear in the output.
   - **Name modifier**: MediaLive assigns a sequential modifier to each output in the output group: **_1**, **_2**, and so on. Change the name if you want.
4. Complete **Stream settings**. This section contains fields for the output encodes (the video, audio, and captions) to create in the output. For information about creating encodes, see the following sections:
   - the section called "Set up video"

- the section called “Set up audio”
- the section called “Set up captions”

Creating a frame capture output group

A *frame capture* output lets you capture the video as a series of files, with each file containing one JPEG image. For example, the output might capture every 10th output frame. You save the files to an S3 bucket.

When you planned the workflow for your channel, you might have determined that you want to include a frame capture output group. A frame capture group always sends output to an S3 bucket.

Topics

- Procedure to create a frame capture output group
- Frame capture destination
- Settings for the stream

Procedure to create a frame capture output group

Follow these steps to create a frame capture output group and output.

To create a Frame Capture output group and its output

1. On the **Create channel** page, under **Output groups**, choose **Add**.
2. In the **Add output group** section, choose **Frame capture**, and then choose **Confirm**. More sections appear.

   - **Destination** – This section contains fields for the output destination.
   - **Frame capture settings** – This section contains a field for the output group name and for the output destination.
   - **Frame capture outputs** – This section shows the output that is added by default. A frame capture output can contain only one output, so don't click **Add output**.

     To view the fields, choose the **Settings** link.
3. In **Frame capture outputs**, choose the **Settings** link to view the sections for the individual output:

- **Output settings** – This section contains fields for the output destination.
- **Stream settings** – This section contains fields for the output streams (the video, audio, and captions).

4. (Optional) Enter names for the output group and the output:
   - In **Frame capture settings**, for **Name**, enter a name for the output group. This name is internal to MediaLive; it doesn't appear in the output. For example, **`Sports Game Thumbnails`**.
   - In **Frame capture outputs**, for **Name**, enter a name for the output. This name is internal to MediaLive; it doesn't appear in the output.
5. To complete the other fields, see the topics listed after this procedure.
6. After you have finished setting up this output group and its single output, you can create another output group (of any type), if your plan requires it. Otherwise, go to the section called "Save channel".

Frame capture destination

The following fields configure the location and names of the frame capture files (the destination).

- **Output group** – **Frame capture group destination** section
- **Output group** – **Frame capture settings** – **CDN settings**

  Output settings – **Name modifier**

You must design the destination path or paths for the output. You must then enter the different portions of the path into the appropriate fields on the console.

Design the path for the output destination

As part of the planning for this output group, you discussed your requirements with the Amazon S3 user. You should already have the following information:

- The bucket names for the output
- Or the full path for the output

To design the path

If you haven't yet designed the destination path or paths, design them now. If you've already designed the paths, go to the section called “Complete the fields on the console”.

- Design the destination path or paths, following this syntax:

  ```
  protocol bucket folders baseFilename nameModifier counter extension
  ```

  For example, for a standard channel:

  ```
  s3ssl://DOC-EXAMPLE-BUCKET1/sports-thumbnails/delivery/
  curling-20180820.00000.jpg
  ```

  ```
  s3ssl://DOC-EXAMPLE-BUCKET1/sports-thumbnails/backup/
  curling-20180820.00000.jpg
  ```

The following table maps each portion in the example to the portion in the syntax.

Portion of the URL	Example	Comment
protocol	s3ssl://	The protocol is always **`s3ssl://`** because the destination for a frame capture output is always an S3 bucket.
bucket portion of the path	DOC-EXAMPLE-BUCKET1	When you planned the workflow for the channel, you should have made sure that the S3 bucket or buckets exist. With MediaLive, the S3 bucket name must not use dot notation. For example, **`mycompany-videos`** is acceptable but **`mycompany.videos`** isn't.

Portion of the URL	Example	Comment
folders portion of the path	sports-thumbnails/delivery/	The folders can be present or not, and can be as long as you want. The folders must always end with a slash.
baseFilename	curling	Don't terminate the file name with a slash.
nameModifier	-20180820	The modifier is optional for an frame capture output.
delimiter before the counter	.	MediaLive automatically inserts this delimiter.
counter	00000	MediaLive automatically generates this counter. Initially, this is a five-digit number starting at 00000, and increasing by 1. So 00000, 00001, 00002 and so on. After 99999, the next number is 100000 (six digits), then 100001, 100002, and so on. Then from 999999 to 1000000 (seven digits), and so on.
dot before the extension	.	MediaLive automatically inserts this dot.
extension	jpg	Always **`jpg`**.

Complete the fields on the console

To specify the location for the output

1. Enter the different portions of the destination in the appropriate fields.

Portion of the destination URL	Field	Example
protocol, bucket, folders, baseFilename	The two **URL** fields in the **Frame capture group destinations** section. The data before the first slash is the bucket name. The data after the last slash is the baseFilename. The data in between is the folders. Specify two destinations when the channel is set up as a standard channel, or one destination when it is set up as a single-pipeline channel.	`s3ssl://DOC-EXAMPLE-BUCKET1/sports-thumbnails/delivery/curling`

Portion of the destination URL	Field	Example
nameModifier	The **Name modifier** field in the **Frame capture outputs** section. If you choose to include a modifier, you can enter a string such as **`-high`**, to indicate a high-resolution output. Or you can enter a variable ID (such as `$dt$`) to ensure that the modifier is different for each file segment. For a list of variable data identifiers, see the section called "Variable data: Supported identifiers".	`$dft$`

2. Leave the **Credentials** section blank in both the **Frame capture group destinations** sections. MediaLive has permission to write to the S3 bucket via the trusted entity. Someone in your organization should have already set up these permissions. For more information, see the section called "Reference: summary of trusted entity access".
3. Complete the **CDN settings** field only if MediaLive must set a canned ACL whenever it sends this output to the Amazon S3 bucket.

   Use of a canned ACL typically only applies if your organization is not the owner of the Amazon S3 bucket. You should have discussed the use of a canned ACL with the bucket owner when you discussed the destination for the output.

Settings for the stream

By default, the output is set up with one video encode. This is the only encode that a frame capture output can contain. Therefore, you can't add audio or captions encodes or more video encodes.

For information about the fields in the video encode, see the section called “Set up video”.

Creating an HLS output group

When you planned the workflow for your channel, you might have determined that you want to include an HLS output group.

Topics

Procedure to create an HLS output group

Follow these steps to create an HLS output group and its outputs.

To create an HLS output group and its outputs

1. On the **Create channel** page, under **Output groups**, choose **Add**.
2. In the **Add output group** section, choose **HLS**, and then choose **Confirm**. More sections appear:

   - **HLS group destination** – This section contains fields for the destination of the outputs. For more information see the section for the type of downstream system:
     - the section called “Destination fields – Amazon S3”
     - the section called “Destination fields – MediaStore”
     - the section called “Destination fields – MediaPackage”

    - the section called “Destination fields – HTTP server”
  - **HLS settings** – This section contains fields for the destination of the outputs, for resiliency, and for captions.
  - **HLS outputs** – This section shows the single output that is added by default.
  - **Location** – This section contains fields for customizing the paths inside the manifests.
  - **Manifest and segments** – This section contains fields for configuring redundant manifests, for configuring the manifest contents, and for configuring media segments.
  - **DRM** – This section contains fields for configuring encryption of outputs.
  - **Ad marker** – This section contains fields for setting up for SCTE-35 ad avails.
  - **Captions** – This section contains fields for configuring captions.
  - **ID3** – This section contains fields for setting up for ID3.
3. If your plan includes more than one output in this output group, then in **HLS outputs**, choose **Add output** to add the appropriate number of outputs.
4. In **HLS outputs**, choose the first **Settings** link to view the sections for the first output:
  - **Output settings** – This section contains fields for the destination of the outputs. See these sections:
    - the section called “Destination fields – Amazon S3”
    - the section called “Destination fields – MediaStore”
    - the section called “Destination fields – MediaPackage”
    - the section called “Destination fields – HTTP server”

    This section also contains fields for the HLS container.
  - **Stream settings** – This section contains fields for the output streams (the video, audio, and captions).
5. (Optional) Enter names for the output group and the outputs:
  - In **HLS settings**, for **Name**, enter a name for the output group. This name is internal to MediaLive; it doesn't appear in the output. For example, **`Sports Curling`**.
  - In the **HLS outputs** section for each output, for **Name**, enter a name for the output. This name is internal to MediaLive; it doesn't appear in the output. For example, **`high resolution`**.
6. To complete the other fields, see the topics listed after this procedure.

7. After you have finished setting up this output group and its outputs, you can create another output group (of any type), if your plan requires it. Otherwise, go to the section called "Save channel".

Fields for the output destination – sending to Amazon S3

When you planned the destinations for the HLS output group, you might have decided to send the output to Amazon S3. You must design the destination path or paths for the output. You must then enter the different portions of the path into the appropriate fields on the console.

Topics

- Step 1: Design the path for the output destination
- Step 2: Complete the fields on the console

Step 1: Design the path for the output destination

Perform this step if you haven't yet designed the full destination path or paths. If you've already designed the paths, go to the section called "Step 2: Complete the fields".

To design the path

1. Collect the bucket names that you previously obtained from the Amazon S3 user. For example:

   ```
   DOC-EXAMPLE-BUCKET
   ```

2. Design the portions of the destination paths that follow the bucket or buckets. For details, see the sections that follow.

Topics

- The syntax for the paths for the outputs
- Designing the folders and baseFilename
- Designing the nameModifier
- Designing the segmentModifier

The syntax for the paths for the outputs

An HLS output always includes three categories of files:

- The main manifest
- The child manifests
- The media files

The following table describes the parts that make up the destination paths for these three categories of files.

The destination paths for these three categories of files are identical up to and including the *baseFilename*, which means that MediaLive sends all these categories of files to the same folder. The modifiers and file extensions are different for each category of file. When sending to Amazon S3, you must send all the files to the same folder. The downstream systems expect all the files to be together.

File	Syntax of the path	Example
Main manifest files	`protocol bucket path baseFilename extension`	The path for a main manifest in the bucket *sports*, with the file name *index*: `s3ssl://DOC-EXAMPLE-BUCKET/sports/delivery/curling/index.m3u8`
Child manifest files	`protocol bucket path baseFilename nameModifier extension`	The path for the child manifest for the high-resolution renditions of the curling output `s3ssl://DOC-EXAMPLE-BUCKET/sports/delivery/curling/index-high.m3u8`
Media files (segments)	`protocol bucket path baseFilename nameModifier`	The path for the file for the 230th segment might be: `s3ssl://DOC-EXAMPLE-BUCKET/sports/de`

File	Syntax of the path	Example
	`optionalSegmentModifier counter extension`	`livery/curling/index-high-00230.ts`

These destination paths are constructed as follows:

- The Amazon S3 user should have provided you with the bucket names.
- You must determine the following:
  - The folders
  - The baseFilename
  - The modifier
  - The segmentModifier

  See the sections that follow.
- MediaLive inserts the underscore before the counter.
- MediaLiveautomatically generates this counter. Initially, this is a five-digit number starting at 00001, and increasing by 1. So 00001, 00002, 00003 and so on. After 99999, the next number is 100000 (six digits), then 100001, 100002, and so on. Then from 999999 to 1000000 (seven digits), and so on.
- MediaLive inserts the dot before the extension.
- MediaLive selects the extension:
  - For manifest files – always `.m3u8`
  - For media files – .ts for files in a transport stream, or .mp4 for files in an fMP4 container

Designing the folders and baseFilename

Design a folder path and baseFilename that suits your purposes.

If you have two destinations for each output, the destination paths must be different from each other in some way. Follow these guidelines:

- At least one of the portions of one path must be different from the other. It is acceptable for all the portions to be different.

Therefore, if the buckets are *different*, the folder path and file names for the two destinations can be different from each other, or they can be the same. For example:

```
s3ssl://DOC-EXAMPLE-BUCKET/sports/delivery/curling/index-high.m3u8
```

```
s3ssl://DOC-EXAMPLE-BUCKET1/sports/delivery/curling/index-high.m3u8
```

or

```
s3ssl://DOC-EXAMPLE-BUCKET/sports/delivery/curling/index-high.m3u8
```

```
s3ssl://DOC-EXAMPLE-BUCKET1/sports/redundant/curling/index-high.m3u8
```

- If the buckets are *the same*, the folder path and file names for the two destinations must be different from each other. For example:

```
s3ssl://DOC-EXAMPLE-BUCKET/sports/delivery/curling/index-high.m3u8
```

```
s3ssl://DOC-EXAMPLE-BUCKET/sports/redundant/curling/index-high.m3u8
```

Designing the nameModifier

Design the `nameModifier` portions of the file name. The child manifests and media files include this modifier in their file names. This `nameModifier` distinguishes each output from the other, so it must be unique in each output. Follow these guidelines:

- For an output that contains video (and possibly other streams), you typically describe the video. For example, **`-high`** or **`-1920x1080-5500kpbs`** (to describe the resolution and the bitrate).
- For an output that contains only audio or only captions, you typically describe the audio or captions. For example, **`-aac`** or **`-webVTT`**.
- It's a good idea to start the `nameModifier` with a delimiter, such as a hyphen, in order to separate the `baseFilename` from the `nameModifier`.
- The `nameModifier` can include data variables.

Designing the segmentModifier

Design the segmentModifiers portion of the destination path. The segmentModifier is optional, and if you include it, only the media file names include it.

A typical use case for this modifier is to use a data variable to create a timestamp, to prevent segments overriding each other if the channel restarts. For example, assume that you include the timestamp **t-**. Segment 00001 might have the name `index-120028-00001`. If the output restarts a few minutes later (which causes the segment counter to restart), the new segment 00001 will have the name `index-120039-00001`. The new file won't overwrite the file for the original segment 00001. Some downstream systems might prefer this behavior.

Step 2: Complete the fields on the console

After you have designed the output names and destination paths, you can set up the HLS output group.

The following fields configure the location and names of the HLS media and manifest files (the destination).

- **Output group – HLS group destination** section
- **Output group – HLS settings – CDN** section
- **Output group – Location – Directory structure**
- **Output group – Location – Segments per subdirectory**
- **HLS outputs – Output settings – Name modifier**
- **HLS outputs – Output settings – Segment modifier**

To set the destination for most downstream systems

1. Complete the **URL** fields in the **HLS group destinations** section. Specify two destinations if the channel is set up as a standard channel, or one destination if it is set up as a single-pipeline channel.

Portion of the destination path	Location of the Field	Description
protocol	**URL** in **HLS group destinations** section	`s3ssl://`

Portion of the destination path	Location of the Field	Description
domain	**URL** in **HLS group destinations** section	The bucket name
path	**URL** in **HLS group destinations** section	The optional path of folders Always terminate with a slash
baseFilename	**URL** in **HLS group destinations** section	Required Don't terminate the baseFilename with a slash.
modifier	**Name modifier** in each **HLS outputs** section	Required Make sure the modifiers are unique across all outputs in the output group
segmentModifier	Segment modifier in each **HLS outputs** section	Optional Keep in mind that this field exists for each output.

2. Leave the **Credentials** section blank in both the **HLS group destinations** sections. MediaLive has permission to write to the S3 bucket via the trusted entity. Someone in your organization should have already set up these permissions. For more information, see the section called "Reference: summary of trusted entity access".
3. In the **CDN** settings section, choose `Hls S3`.
4. Complete the **CDN settings** field only if MediaLive must set a canned ACL whenever it sends this output to the Amazon S3 bucket.

   Use of a canned ACL typically only applies if your organization is not the owner of the Amazon S3 bucket. You should have discussed the use of a canned ACL with the bucket owner when you discussed the destination for the output.

Fields for the output destination – sending to MediaStore

When you planned the destinations for the HLS output group, you might have decided to send the output to MediaStore. You must design the destination path or paths for the output. You must then enter the different portions of the path into the appropriate fields on the console.

Topics

Step 1: Design the path for the output destination

Perform this step if you haven't yet designed the full destination path or paths. If you've already designed the paths, go to the section called “Step 2: Complete the fields”.

To design the path

1. Collect the data endpoint for the container or containers. You previously obtained this information from the MediaStore user. For example:

   ```
   a23f.data.mediastore.us-west-2.amazonaws.com
   ```

2. Design the portions of the destination paths that follow the data endpoint (for MediaStore).

Topics

The syntax for the paths for the outputs

An HLS output always includes three categories of files:

- The main manifest
- The child manifests

- The media files

The following table describes the parts that make up the destination paths for these three categories of files.

The destination paths for these three categories of files are identical up to and including the *baseFilename*, which means that MediaLive sends all these categories of files to the same folder. The modifiers and file extensions are different for each category of file. When sending to MediaStore, you must send all the files to the same folder. The downstream systems expect all the files to be together.

File	Syntax of the path	Example
Main manifest files	`protocol dataEndpoint path baseFilename extension`	The path for a main manifest in the path *delivery* in the container, and with the file name *index*: `mediastoressl://a23f.data.mediastore.us-west-2.amazonaws.com/delivery/index.m3u8`
Child manifest files	`protocol dataEndpoint path baseFilename nameModifier extension`	The path for the child manifest for the high-resolution renditions of the output `mediastoressl://a23f.data.mediastore.us-west-2.amazonaws.com/delivery/index-high.m3u8`
Media files (segments)	`protocol dataEndpoint path baseFilename nameModifier`	The path for the file for the 230th segment might be: `mediastoressl://a23f.data.mediastore`

File	Syntax of the path	Example
	`optionalSegmentModifier counter extension`	`.us-west-2.amazonaws.com/delivery/index-high-00230.ts`

How MediaLive constructs the paths

These paths are constructed as follows:

- The user of the AWS service should have provided you with the container names.
- For MediaStore, you must determine the following:
  - The folders
  - The baseFilename
  - The modifier
  - The segmentModifier

  See the sections that follow.
- MediaLive inserts the underscore before the counter.
- MediaLive generates the counter, which is always five digits starting at 00001.
- MediaLive inserts the dot before the extension.
- MediaLive selects the extension:
  - For manifest files – always `.m3u8`
  - For media files – .ts for files in a transport stream, or .mp4 for files in an fMP4 container

Designing the folders and baseFilename

Design a folder path and baseFilename that suits your purposes.

If you have two destinations for each output, the destination paths must be different from each other in some way. Follow these guidelines:

- At least one of the portions of one path must be different from the other. It is acceptable for all the portions to be different.

Therefore, if the buckets or containers are different, the folder path and file names for the two destinations can be different from each other, or they can be the same. For example:

```
mediastoressl://a23f.data.mediastore.us-west-2.amazonaws.com/delivery/
index.m3u8
```

```
mediastoressl://fe30.data.mediastore.us-west-2.amazonaws.com/delivery/
index.m3u8
```

or

```
mediastoressl://a23f.data.mediastore.us-west-2.amazonaws.com/delivery/
index.m3u8
```

```
mediastoressl://fe30.data.mediastore.us-west-2.amazonaws.com/redundant/
index.m3u8
```

- If the buckets or containers are the same, the folder path and file names for the two destinations must be different from each other. For example:

```
mediastoressl://a23f.data.mediastore.us-west-2.amazonaws.com/delivery/
index.m3u8
```

```
mediastoressl://a23f.data.mediastore.us-west-2.amazonaws.com/redundant/
index.m3u8
```

Designing the nameModifier

Design the `nameModifier` portions of the file name. The child manifests and media files include this modifier in their file names. This `nameModifier` distinguishes each output from the other, so it must be unique in each output. Follow these guidelines:

- For an output that contains video (and possibly other streams), you typically describe the video. For example, **`-high`** or **`-1920x1080-5500kpbs`** (to describe the resolution and the bitrate).
- For an output that contains only audio or only captions, you typically describe the audio or captions. For example, **`-aac`** or **`-webVTT`**.
- It's a good idea to start the `nameModifier` with a delimiter, such as a hyphen, in order to separate the `baseFilename` from the `nameModifier`.
- The `nameModifier` can include data variables.

Designing the segmentModifier

Design the segmentModifiers portion of the destination path. The segmentModifier is optional, and if you include it, only the media file names include it.

A typical use case for this modifier is to use a data variable to create a timestamp, to prevent segments overriding each other if the channel restarts. For example, assume that you include the timestamp **t-**. Segment 00001 might have the name `index-120028-00001`. If the output restarts a few minutes later (which causes the segment counter to restart), the new segment 00001 will have the name `index-120039-00001`. The new file won't overwrite the file for the original segment 00001. Some downstream systems might prefer this behavior.

Step 2: Complete the fields on the console

After you have designed the output names and destination paths, you can set up the HLS output group.

The following fields configure the location and names of the HLS media and manifest files (the destination).

- **Output group – HLS group destination** section
- **Output group – HLS settings – CDN** section
- **Output group – Location – Directory structure**
- **Output group – Location – Segments per subdirectory**
- **HLS outputs – Output settings – Name modifier**
- **HLS outputs – Output settings – Segment modifier**

To set the destination for most downstream systems

1. Complete the **URL** fields in the **HLS group destinations** section. Specify two destinations if the channel is set up as a standard channel, or one destination if it is set up as a single-pipeline channel.

Portion of the destination path	Location of the Field	Description
protocol	**URL** in **HLS group destinations** section	`mediastoressl://`
domain	**URL** in **HLS group destinations** section	The data endpoint
path	**URL** in **HLS group destinations** section	The optional path of folders Always terminate with a slash
baseFilename	**URL** in **HLS group destinations** section	Required Don't terminate the baseFilename with a slash.
modifier	**Name modifier** in each **HLS outputs** section	Required Make sure the modifiers are unique across all outputs in the output group
segmentModifier	Segment modifier in each **HLS outputs** section	Optional Keep in mind that this field exists for each output.

2. Leave the **Credentials** section blank in both the **HLS group destinations** sections. MediaLive has permission to write to the MediaStore container via the trusted entity. Someone in your organization should have already set up these permissions. For more information, see the section called “Reference: summary of trusted entity access”.
3. In the **CDN** settings section, choose `Hls media store`.
4. If the MediaStore user gave you values to configure the connection, enter those values in the fields in the **CDN** settings section.

Fields for the output destination – sending to MediaPackage

When you planned the output to MediaPackage, you might have decided to send the output by creating an HLS output group. (Or you might have decided to create a MediaPackage output group.)

You must design the destination path or paths for the output. You must then enter the different portions of the path into the appropriate fields on the console.

You can use an HLS output group to send to standard MediaPackage or toMediaPackage v2. The two versions use different protocols:

- MediaPackage uses WebDAV.
- MediaPackage v2 uses Basic PUT.

Topics

Step 1: Design the path for the output destination

Perform this step if you haven't yet designed the full destination path or paths. If you've already designed the paths, go to the section called “Step 2: Complete the fields”.

To design the path

1. Collect the information you previously obtained from the MediaPackage user:
   - The two URLs (input endpoints is the MediaPackage terminology) for the channel. See the information after this procedure.
   - If you are using standard MediaPackage, obtain the user name and password. If you are using MediaPackage v2, you don't use user credentials.
2. You must design the portions of the destination paths that follow the URLs.

Topics

- Collect the information for standard MediaPackage
- Collect the information for MediaPackage v2
- The syntax for the paths for the outputs
- Designing the nameModifier
- Designing the segmentModifier

Collect the information for standard MediaPackage

For standard MediaPackage, the two URLs for a channel look like these examples:

`6d2c.mediapackage.us-west-2.amazonaws.com/in/v2/9dj8/9dj8/channel`

`6d2c.mediapackage.us-west-2.amazonaws.com/in/v2/9dj8/e333/channel`

Where:

`mediapackage` indicates that the input endpoints uses version 1 of the MediaPackage API

`channel` always appears at the end of the URL. It is the base filename for all the files for this destination.

The two URLs are always identical except for the folder just before `channel`.

Collect the information for MediaPackage v2

For MediaPackage v2, the two URLs for a channel look like these examples:

`mz82o4-1.ingest.hnycui.mediapackagev2.us-west-2.amazonaws.com/in/v1/live-sports/1/curling/index`

`mz82o4-2.ingest.hnycui.mediapackagev2.us-west-2.amazonaws.com/in/v1/live-sports/2/curling/index`

Where:

Element	Description
`mz82o4-1` and `mz82o4-2`	Indicate that the two endpoints are for a redundant channel in MediaPackage. The prefixes are always `-1` and `-2`

Element	Description
`mediapackagev2`	Indicates that the input endpoints uses version 2 of the MediaPackage API
`live-sports/1/curling` and `live-sports/2/curling`	Folders for the redundant ingests. One folder always includes `/1/`, and the other folder always includes `/2/`
`index`	Always appears at the end of the URL. It is the base filename for all the files for this destination.

The syntax for the paths for the outputs

An HLS output always includes three categories of files:

See the following sections.

- The main manifest
- The child manifests
- The media files

The following table describes the parts that make up the destination paths for these three categories of files.

The destination paths for these three categories of files are identical up to and including the *baseFilename*, which means thatMediaLive sends all these categories of files to the same folder. The modifiers and file extensions are different for each category of file. When sending to MediaPackage, you must send all the files to the same folder. The downstream systems expect all the files to be together.

File	Syntax of the path	Example
Main manifest files	`protocol channelURL extension`	The path for output. Here is an example that uses MediaPackage v2

File	Syntax of the path	Example
		`https://mz82o4-2.ingest.hnycui.mediapackagev2.us-west-2.amazonaws.com/in/v1/live-sports/2/curling/index.m3u8`
Child manifest files	`protocol channelURL nameModifier extension`	Here is an example for the path for the child manifest for the high-resolution renditions of the curling output (in a destination that uses MediaPackage v2): `https://mz82o4-1.ingest.hnycui.mediapackagev2.us-west-2.amazonaws.com/in/v1/live-sports/1/curling/index-high.m3u8`
Media files (segments)	`protocol channelURL nameModifier optionalSegmentModifier counter extension`	Here is an example for the path for the file for the 230th segment (in a destination that uses MediaPackage v2): `https://mz82o4-1.ingest.hnycui.mediapackagev2.us-west-2.amazonaws.com/in/v1/live-sports/1/curling/index-high-00230.ts`

These paths are constructed as follows:

- The MediaPackage user should have provided you with the channel URLs. The URLs cover the portion of the path up to and including the baseFilename:
  - With standard MediaPackage, the baseFilename is always `channel`.
  - With MediaPackage v2, the baseFilename is always `index`.
- You must specify the following:
  - The modifier
  - The segmentModifier

  See the sections that follow.
- MediaLive inserts the underscore before the counter.
- MediaLive generates the counter, which is always five digits starting at 00001.
- MediaLive inserts the dot before the extension.
- MediaLive selects the extension:
  - For manifest files – always `.m3u8`
  - For media files – .ts for files in a transport stream, or .mp4 for files in an fMP4 container

Designing the nameModifier

Design the `nameModifier` portions of the file name. The child manifests and media files include this modifier in their file names.

This `nameModifier` distinguishes each output from the other, so it must be unique in each output.

- For an output that contains video (and possibly other streams), you typically describe the video. For example, if you have three renditions, you might use **`-high`**, **`-medium`** and **`-low`**. Or each modifier could accurately describe the resolution and the bitrate (**`-1920x1080-5500kpbs`**).
- For an output that contains only audio or only captions, you typically describe the audio or captions. For example, **`-aac`** or **`-webVTT`**.

It's a good idea to start the `nameModifier` with a delimiter, such as a hyphen, in order to separate the `baseFilename` from the `nameModifier`.

The `nameModifier` can include data variables.

Designing the segmentModifier

Design the segmentModifiers portion of the destination path. The segmentModifier is optional, and if you include it, only the media file names include it.

A typical use case for this modifier is to use a data variable to create a timestamp, to prevent segments overriding each other if the channel restarts. For example, assume that you include the timestamp **t-**. Segment 00001 might have the name `index-120028-00001`. If the output restarts a few minutes later (which causes the segment counter to restart), the new segment 00001 will have the name `index-120039-00001`. The new file won't overwrite the file for the original segment 00001. Some downstream systems might prefer this behavior.

Step 2: Complete the fields on the console

After you have designed the output names and destination paths, you can set up the HLS output group.

The following fields configure the location and names of the HLS media and manifest files (the destination).

- **Output group – HLS group destination** section
- **Output group – HLS settings – CDN** section
- **Output group – Location – Directory structure**
- **Output group – Location – Segments per subdirectory**
- **HLS outputs – Output settings – Name modifier**
- **HLS outputs – Output settings – Segment modifier**

To set the destination

1. Complete the **URL** fields in the **HLS group destinations** section. Specify two destinations if the channel is set up as a standard channel, or one destination if it is set up as a single-pipeline channel.

Portion of the destination path	Location of the Field	Description
protocol	**URL** in **HLS group destinations** section	Enter `https://`
domain	**URL** in **HLS group destinations** section	Enter the MediaPackage channel URL
path	**URL** in **HLS group destinations** section	Not applicable, the path is already specified in the channel URL
baseFilename	**URL** in **HLS group destinations** section	Not applicable, the path is already specified in the channel URL With MediaPackage, the `baseFilename` is always **`channel`**. With MediaPackage v2 it is always **`index`**. Don't terminate the **baseFilename** with a slash.
modifier	**Name modifier** in each **HLS outputs** section	Required. For guidance, see the section called "Name modifier". Make sure the modifiers are unique across all outputs in the output group
segmentModifier	Segment modifier in each **HLS outputs** section	Optional. For guidance, see the section called "Segment modifier". Keep in mind that this field exists for each output.

2. Enter the input user name. For the password (if applicable), enter the name of the password stored on the AWS Systems Manager Parameter Store. Don't enter the password itself. For more information, see the section called "AWS Systems Manager parameter store".

3. In the **CDN** settings section, choose the appropriate connection type:
   - To send to standard MediaPackage, choose `Hls webdav`.
   - To send to MediaPackage v2, choose `Basic PUT`.
4. If the downstream system gave you values to configure the connection, enter those values in the fields in the **CDN** settings section.

Standard MediaPackage example

This example shows how to set up the destination fields if the downstream system for the HLS output group is standard MediaPackage.

Assume that you want to stream the curling game and to create three outputs: high, medium, and low bitrate.

Field	Value
CDN settings in **HLS settings** section	**`hls webdav`**
URL in **HLS group destination A** section	**`6d2c.mediapackage.us-west-2.amazonaws.com/in/v2/9dj8/9dj8/channel`**
Credentials in **HLS group destination A** section	MediaPackage accepts only authenticated requests, so you must enter a user name and a password that is known to MediaPack age. For the password, enter the name of the password stored on the AWS Systems Manager Parameter Store. Don't enter the password itself. For more information, see the section called “AWS Systems Manager parameter store”.
URL in **HLS group destination B** section	**`6d2c.mediapackage.us-west-2.amazonaws.com/in/v2/9dj8/e333/channel`**

Field	Value
Credentials in **HLS group destination B** section	Enter a user name and password for the URL for destination B. The credentials are probably the same for both URLs, but they might not be.
Name modifier in **HLS outputs** section	Choose **Add output** twice: two more **Output** lines are added to this section, for a total of three lines. In each line, enter a modifier: **`-high`**, **`-medium`**, and **`-low`**.
Directory Structure and **Segments Per Subdirectory** in **Location** section	MediaPackage doesn't use these fields, therefore leave them blank.

As a result, files are created with the following names:

- One main manifest: **`channel.m3u8`**
- One child manifest for each output: **`channel-high.m3u8`**, **`channel-medium.m3u8`**, **`channel-low.m3u8`**
- TS files for each output:
  - **`channel-high-00001.ts`**, **`channel-high-00002.ts`**, **`channel-high-00003.ts`**, and so on
  - **`channel-medium-00001.ts`**, **`channel-medium-00002.ts`**, **`channel-medium-00003.ts`**, and so on
  - **`channel-low-00001.ts`**, **`channel-low-00002.ts`**, **`channel-low-00003.ts`**, and so on

The files will be published to both URL inputs on MediaPackage.

MediaPackage v2 example

This example shows how to set up the destination fields if the downstream system for the HLS output group is standard MediaPackage.

Assume that you want to stream the curling game and to create three outputs: high, medium, and low bitrate.

Field	Value
CDN settings in **HLS settings** section	**`basic PUT`**
URL in **HLS group destination A** section	**`mz82o4-1.ingest.hnycui.mediapackagev2.us-west-2.amazonaws.com/in/v1/live-sports/1/curling/index`**
Credentials in **HLS group destination A** section	Leave blank. MediaPackage v2 doesn't use credentials to authenticate.
URL in **HLS group destination B** section	**`mz82o4-2.ingest.hnycui.mediapackagev2.us-west-2.amazonaws.com/in/v1/live-sports/2/curling/index`** .
Credentials in **HLS group destination B** section	Leave blank. MediaPackage v2 doesn't use credentials to authenticate.
Name modifier in **HLS outputs** section	Choose **Add output** twice: two more **Output** lines are added to this section, for a total of three lines. In each line, enter a modifier: **`-high`**, **`-medium`**, and **`-low`**.
Directory Structure and **Segments Per Subdirectory** in **Location** section	MediaPackage doesn't use these fields, therefore leave them blank.

As a result, files are created with the following names:

- One main manifest: **`index.m3u8`**
- One child manifest for each output: **`index-high.m3u8`**, **`index-medium.m3u8`**, **`index-low.m3u8`**
- TS files for each output:
  - **`index-high-00001.ts`**, **`index-high-00002.ts`**, **`index-high-00003.ts`**, and so on
  - **`index-medium-00001.ts`**, **`index-medium-00002.ts`**, **`index-medium-00003.ts`**, and so on

- **`index-low-00001.ts`**, **`index-low-00002.ts`**, **`index-low-00003.ts`**, and so on

The files will be published to both URL inputs on MediaPackage.

Fields for the output destination – sending to an HTTP server

When you planned the destinations for the HLS output group, you might have decided to send the output to an HTTP server.

You must design the destination path or paths for the output. You must then enter the different portions of the path into the appropriate fields on the console.

Topics

Step 1: Design the path for the output destination

Perform this step if you haven't yet designed the full destination path or paths. If you've already designed the paths, go to the section called “Step 2: Complete the fields”.

To design the path

1. Collect the information that you previously obtained from the operator of the downstream system:

   - The connection type for the downstream system – Akamai, basic PUT, or WebDAV.
   - The settings for connection fields, if the downstream system has special requirements.
   - The protocol for delivery—HTTP or HTTPS.
   - The user name and password to access the downstream system, if the downstream system requires authenticated requests. Note that these user credentials relate to user authentication, not to the protocol. User authentication is about whether the downstream system will accept your request. The protocol is about whether the request is sent over a secure connection.

- All or part of the destination paths, possibly including the file names.
- Whether you need to set up separate subdirectories.

2. As part of the planning with the operator of the downstream system, you should have determined if you want to implement redundant manifests. You should also have determined if the downstream system requires custom manifests. Given these two decisions, read the appropriate section:

   - If you are implementing redundant manifests, see the section called "Manifests – Redundant HLS manifests", then return to this section.
   - If you are implementing custom paths for manifests, see the section called "Manifests – custom HLS manifest paths", then return to this section.
   - If you are not implementing either of those features, continue keep reading this section.

3. Design the portions of the destination paths that follow the bucket or buckets. For details, see the sections that follow.

Topics

The syntax for the paths for the outputs

The following table describes the parts that make up the destination paths for these three categories of files.

The destination paths for these three categories of files are identical up to and including the *baseFilename*, which means thatMediaLive sends all these categories of files to the same folder. The modifiers and file extensions are different for each category of file.

File	Syntax of the path	Example
Main manifest files	protocol domain path baseFilename extension	The URL for a main manifest with the file name */index*:

File	Syntax of the path	Example
		`http://203.0.113.55/sports/delivery/curling/index.m3u8`
Child manifest files	protocol domain path baseFilename nameModifier extension	The URL for the child manifest for the high-resolution renditions of the output `http://203.0.113.55/sports/delivery/curling/index-high.m3u8`
Media files (segments)	`protocol domain path baseFilename nameModifier optionalSegmentModifier counter extension`	The URL for the file for the 230th segment might be: `http:// 203.0.113.55/sports/delivery/curling/index-high-00230.ts`

These destination paths are constructed as follows:

- The operator at the downstream system should have provided you with the protocol, domain and part of the path. For example:

  `http://203.0.113.55/sports/`

  The protocol is always HTTP or HTTPS.
- The operator might have provided the following. Otherwise, you decide them:
  - The folders
  - The baseFilename
  - The modifier
  - The segmentModifier

See the sections that follow.

- MediaLive inserts the underscore before the counter.
- MediaLive generates the counter, which is always five digits starting at 00001.
- MediaLive inserts the dot before the extension.
- MediaLive selects the extension:
  - For manifest files – always `.m3u8`
  - For media files – `.ts` for files in a transport stream, and `.mp4` for files in an fMP4 container

Designing the folders and baseFilename

For the `folder` and `baseFilename` portion of the destination path, follow these guidelines:

- For a single-pipeline channel, you need only one `baseFilename`.
- For a standard channel when you are *not* implementing redundant manifests, you need two `baseFilenames`. The two `baseFilenames` can be identical or different. Before you create different `baseFilenames`, make sure that the downstream system can work with that setup.
- For a standard channel when you *are* implementing redundant manifests, see the section called "Redundant manifest fields".

Designing the nameModifier

Design the `nameModifier` portions of the file name. The child manifests and media files include this modifier in their file names. This `nameModifier` distinguishes each output from the other, so it must be unique in each output. Follow these guidelines:

- For an output that contains video (and possibly other streams), you typically describe the video. For example, **`-high`** or **`-1920x1080-5500kpbs`** (to describe the resolution and the bitrate).
- For an output that contains only audio or only captions, you typically describe the audio or captions. For example, **`-aac`** or **`-webVTT`**.
- It's a good idea to include a delimiter, to clearly separate the `baseFilename` from the `nameModifier`.
- The `nameModifier` can include data variables.

Designing the segmentModifier

Design the segmentModifiers portion of the destination path. The segmentModifier is optional, and if you include it, only the media file names include it.

A typical use case for this modifier is to use a data variable to create a timestamp, to prevent segments overriding each other if the channel restarts. For example, assume that you include the timestamp **t-**. Segment 00001 might have the name `/index-120028-00001`. If the output restarts a few minutes later (which causes the segment counter to restart), the new segment 00001 will have the name `/index-120039-00001`. The new file won't overwrite the file for the original segment 00001. Some downstream systems might prefer this behavior.

Step 2: Complete the fields on the console

The following fields configure the location and names of the HLS media and manifest files (the destination).

- **Output group – HLS group destination** section
- **Output group – HLS settings – CDN** section
- **Output group – Location – Directory structure**
- **Output group – Location – Segments per subdirectory**
- **HLS outputs – Output settings – Name modifier**
- **HLS outputs – Output settings – Segment modifier**

To set the destination

1. Complete the **URL** fields in the **HLS group destinations** section. Specify two destinations if the channel is set up as a standard channel, or one destination if it is set up as a single-pipeline channel.

Portion of the destination URL	Location of the Field	Example
protocol	**URL** in **HLS group destinations** section	**`http://`**

Portion of the destination URL	Location of the Field	Example
domain	**URL** in **HLS group destinations** section	`203.0.113.55`
path	**URL** in **HLS group destinations** section	`/sports/delivery/curling/` Always terminate with a slash
baseFilename	**URL** in **HLS group destinations** section	`index` Don't terminate the baseFilename with a slash.
modifier	**Name modifier** in each **HLS outputs** section	Required Make sure the modifiers are unique across all outputs in the output group
segmentModifier	Segment modifier in each **HLS outputs** section	Optional Keep in mind that this field exists for each output.

2. If the downstream system requires user authentication from MediaLive, in each **HLS group destination** section, complete the **Credentials** section. Enter a user name and a password provided by the downstream system. For the password, enter the name of the password stored on the AWS Systems Manager Parameter Store. Don't enter the password itself. For more information, see the section called “AWS Systems Manager parameter store”.
3. In the **CDN** settings section, choose the option that the downstream system told you to use—Akamai, PUT, or WebDAV.
4. If the downstream system gave you values to configure the connection, enter those values in the fields in the **CDN** settings section.

Example for an HTTP or HTTPS server

This example shows how to set up the destination fields if the downstream system is an HTTPS server that uses basic PUT.

Assume that you want to stream the curling game and to create three outputs: high, medium, and low bitrate.

Field	Value
CDN settings in **HLS settings** section	**`Hls basic put`** Change the other CDN fields according to the instructions from the downstream system.
URL in **HLS group destination A** section	For example: **`https://203.0.113.55/sports/curling/index`**
Credentials in **HLS group destination A** section	If the downstream system requires authentic ated requests, enter the user name provided by the downstream system. For the password, enter the name of the password stored on the AWS Systems Manager Parameter Store. Don't enter the password itself. For more informati on, see the section called “AWS Systems Manager parameter store”.
URL in **HLS group destination B** section	For example: **`https://203.0.113.82/sports/curling/index`**
Credentials in **HLS group destination B** section	Enter a user name and password for the URL for destination B, if applicable. The credentials are probably the same for both URLs, but they might not be.

Field	Value
Name modifier in **HLS outputs** section	Choose **Add output** twice: two more **Output** lines are added to this section, for a total of three lines. In each line, enter a modifier: **`-high`**, **`-medium`**, and **`-low`**.
Directory Structure and **Segments Per Subdirectory** in **Location** section	Assume that the downstream system doesn't use these fields.

As a result, files are created with the following names:

- One main manifest: `index.m3u8`
- One child manifest for each output: `index-high.m3u8`, `index-medium.m3u8`, `index-low.m3u8`
- TS files for each output:
  - `index-high-00001.ts`, `index-high-00002.ts`, `index-high-00003.ts`, and so on
  - `index-medium-00001.ts`, `index-medium-00002.ts`, `index-medium-00003.ts`, and so on
  - `index-low-00001.ts`, `index-low-00002.ts`, `index-low-00003.ts`, and so on

The files will be published to two hosts at the downstream system, and in a folder called `sports` on each host.

Akamai example

This example shows how to set up the destination fields if the downstream system is an Akamai server.

Assume that you want to stream the curling game and to create three outputs: high, medium, and low bitrate.

Field	Value
CDN settings in **HLS settings** section	**`HLS akamai`**

Field	Value
	Select this setting if you are using Akamai Token Authentication. Change the other CDN fields according to the instructions from Akamai. **`HLS basic put`** Select this setting if you are using digest authentication. Change the other CDN fields according to the instructions from Akamai.
URL in **HLS group destination A** section	For example: **`https://p-ep50002.i.akamaientrypoint.net/50002/curling/index`** Mapping this URL to the Akamai terminology: • *p-ep* stands for primary entry point *https://p-ep50002.i.akamaientrypoint.net* is the hostname • *50002* is the stream ID for the primary entry point • *curling* is the event name • *index* is the manifest name
Credentials in **HLS group destination A** section	If Akamai requires authenticated requests, enter a user name and a password that is known to Akamai. For the password, enter the name of the password stored on the AWS Systems Manager Parameter Store. Don't enter the password itself. For more informati on, see the section called “AWS Systems Manager parameter store”.

Field	Value
URL in **HLS group destination B** section	For example: **`https://b-ep50002.i.akamaientrypoint.net/50002-b/curling/index`** Mapping this URL to the Akamai terminology: • *b-ep* stands for backup entry point *https://b-ep50002.i.akamaientrypoint.net* is the hostname • *50002-b* is the stream ID for the backup entry point • *curling* is the event name • *index* is the manifest name
Credentials in **HLS group destination B** section	Enter a user name and password for the URL for the other destination, if applicable. The credentials are probably the same for both URLs, but they might not be.
Name modifier in **HLS outputs** section	Choose **Add output** twice: two more **Output** lines are added to this section, for a total of three lines. In each line, enter a modifier: **`-high`**, **`-medium`**, and **`-low`**.
Directory Structure and **Segments Per Subdirectory** in **Location** section	Complete the fields according to the instructions from Akamai.

As a result, files are created with the following names:

- One main manifest: **`index.m3u8`**
- One child manifest for each output: **`index-high.m3u8`**, **`index-medium.m3u8`**, **`index-low.m3u8`**

- TS files for each output:
  - `index-high-00001.ts`, `index-high-00002.ts`, `index-high-00003.ts`, and so on
  - `index-medium-00001.ts`, `index-medium-00002.ts`, `index-medium-00003.ts`, and so on
  - `index-low-00001.ts`, `index-low-00002.ts`, `index-low-00003.ts`, and so on

The files will be published to two places:

- On the Akamai host **`p-ep50002.i.akamaientrypoint.net`** in a folder called **`50002`**
- On the host **`b-ep50002.i.akamaientrypoint.net`** in a folder called **`50002-b`**

Fields for the HLS container

The following fields configure the container in each output.

- **HLS outputs** – **Output settings** – **HLS settings** section

These fields control the content of the manifest and structure of the segments. By comparison, fields described in the section called “Manifest content fields” control how many manifests and segments are in the output.

To configure the container

1. In **HLS Settings**, choose the appropriate option. For information on the options, see the list after this procedure.
2. For **Standard hls**, more fields appear. Choose **Transport/container configuration** and **PID settings**. More fields appear.
3. Change any fields. Typically, you change the fields in these two sections only if the downstream system provides you with values.

About HLS containers

MediaLive supports these types of containers:

- **Standard hls** – Choose this type of container if you want to package the streams (encodes) in a transport stream (TS). Choose this container type for all the outputs in the output group (except for outputs that are part of an audio rendition group). Each output might contain these encodes:
  - One video encode
  - One video encode with embedded captions
  - One video encode (and optionally embedded captions) and one or more audio encodes
  - One captions encode
- **Fmp4 hls** – Choose this type of container if you want to package the streams (encodes) as fragmented MP4. Choose this container type for all the outputs in the output group (except for outputs that are part of an audio rendition group). Each output might contain these encodes:
  - One video encode
  - One video encode with embedded captions
  - One captions encode
- **Audio-only** – Choose this type of container for each audio-only output that is part of an audio rendition group. The rendition group can be part of a TS (transport stream) or part of an fMP4 package. For information about creating an audio rendition group, see the section called "Audio – audio rendition groups for HLS".
- **Frame capture** – Choose this type of container to create a JPEG file of frame captures in the output group. This container is used to implement trick-play. For more information about this feature and for instructions on setting it up in the channel, see the section called "Trick-play track via the Image Media Playlist specification".

Fields for customizing the paths inside the manifests

Inside the main manifest, there are paths to each child manifest. Inside each child manifest, there are paths to the media files for that manifest.

You can optionally change the syntax of these paths. Typically, you only need to change the syntax if the downstream system has special path requirements.

The following fields relate to custom paths inside the manifests:

- **HLS output group – Location** – the **Base URL content** fields.
- **HLS output group – Location** – the **Base URL manifest** fields.

For more information about setting up custom paths in manifests, see the section called "Manifests – custom HLS manifest paths".

Fields for redundant manifests

MediaLive supports redundant manifests as specified in the HLS specification. You can enable this feature in a standard channel.

The following fields relate to redundant manifests:

- **HLS output group – Manifests and Segments – Redundant manifests** field
- **HLS output group – Location – the Base URL manifest** fields
- **HLS output group – Location – the Base URL content** fields

You can't enable this feature in an HLS output group that has MediaPackage as the downstream system.

For more information about setting up for redundant manifests, see the section called "Manifests – Redundant HLS manifests".

Fields for the video, audio, and captions streams (encodes)

The following fields relate to the encoding of the video, audio, and captions encodes in each output.

- **Stream settings** section

For information about creating encodes, see the following sections:

- the section called "Set up video"
- the section called "Set up audio"
- the section called "Set up captions"

Fields for other HLS features

Topics

- Fields for connection retries

- Fields for contents of manifests
- Fields for segments
- Fields for resiliency
- Fields for DRM
- Fields for SCTE-35 ad avails
- Fields for captions
- Fields for ID3 metadata

Fields for connection retries

The following fields in the **Output group – HLS settings – CDN settings** section configure the behavior for reconnecting to the downstream system:

- **Connection retry interval**
- **Num retries**
- **Filecache duration**
- **Restart delay**

For details about a field, choose the **Info** link next to the field in the MediaLive console.

Fields for contents of manifests

The following fields in the **HLS output group – Manifests and Segments** section configure the information to include in the HLS child manifests:

- **Output selection**
- **Mode**
- **Stream inf resolution**
- **Manifest duration format**
- **Num segments**
- **I-frame only playlists** – This field is used to implement trick-play via I-frames. For more information, see the section called “Trick-play track via I-frames”.
- **Program date time (PDT)** – This field is used to include or exclude the `EXT-X-PROGRAM-DATE-TIME` tag in manifest files. The tag information helps downstream players to synchronize the stream to the source that's selected in the **PDT clock** field.

- **Program date time (PDT) period** – This field is used to set the time interval for insertion of `EXT-X-PROGRAM-DATE-TIME` tags, in seconds.
- **Program date time (PDT) clock** – This field is used to select the time source of the PDT. Output timecode or UTC time can be selected.
- **Client cache**
- **Timestamp delta microseconds**
- **Codec specification**
- **Manifest compression**

For details about a field, choose the **Info** link next to the field in the MediaLive console.

Fields for segments

The following fields configure media segments in the output.

- The following fields in the **HLS output group – Manifests and Segments** section:
  - **TS file mode**
  - **Segment length**
  - **Keep segments**
  - **Min segment length**
- **HLS outputs** – **Output settings** – **H.265 Packaging type**. This field applies only to fMP4 outputs. MediaLive ignores the value in this field for other types.

For details about a field, choose the **Info** link next to the field.

Fields for resiliency

The following field relates to implementing resiliency in an HLS output.

- **HLS output group** – **HLS Settings** section – **Input loss action**

Optionally change the value of **Input loss action**.

Setting up for most downstream systems

If you're sending this HLS output to a downstream system other than AWS Elemental MediaPackage, choose the **Info** link to decide which option to choose.

Setting up for MediaPackage

If you're sending this HLS output to AWS Elemental MediaPackage, set this field to match how you set the channel class:

- If the channel is a standard channel (to support input redundancy on MediaPackage), set this field to **PAUSE_OUTPUT**.

  With this setup, if MediaLive stops producing output on one pipeline, MediaPackage detects the lack of content on its current input and switches to the other input. Content loss is minimized.

  (If you set this field to **EMIT_OUTPUT**, MediaLive sends filler frames to MediaPackage. MediaPackage doesn't consider filler frames to be lost content, and therefore doesn't switch to its other input.)
- If the channel is a single-pipeline channel, set this field to **EMIT_OUTPUT**.

  With this setup, if the pipeline fails in MediaLive then MediaPackage continues delivering to its own downstream system (although the content will be filler frames).

  (If you set this field to **PAUSE_OUTPUT**, MediaPackage stops updating its endpoint, which might cause problems at the downstream system.)

Fields for DRM

Complete the **DRM** section only if you are setting up for DRM using a static key to encrypt the output.

- In **Key provider** settings, choose **Static key**.
- Complete the other fields as appropriate. For details about a field, choose the **Info** link next to the field.

In a static key setup, you enter an encryption key in this section (along with other configuration data) and then give that key to the other party (for example, by sending it in an email). A static key is not really a DRM solution and is not highly secure.

MediaLive supports only a static key as an encryption option. To use a DRM solution with a key provider, you must deliver the output to AWS Elemental MediaPackage, by creating a MediaPackage output group instead of an HLS output group. You then encrypt the video using MediaPackage. For more information, see the AWS Elemental MediaPackage User Guide.

Fields for SCTE-35 ad avails

Complete the **Ad markers** section if you plan to include SCTE-35 ad messages in the output and to decorate the HLS manifest. See the section called "SCTE-35 message processing" and specifically the section called "Enabling decoration – HLS".

Fields for captions

The following fields relate to embedded captions in an HLS output. If your plan includes creating at least one embedded captions encode in this HLS output, then the following fields apply:

- In the **Captions** section, the **Caption language setting**.

  You can optionally set up the HLS manifest to include information about the languages of the embedded captions.

- **HLS settings** section – **Caption language mappings**

  You can optionally set up the HLS manifest to include information about each CC (caption channel) number and language.

For detailed instructions about both these fields, see the section called "Language information in HLS manifests".

Fields for ID3 metadata

Complete the **ID3** section if you want to insert timed ID3 metadata or ID3 segment tags into all the outputs in this output group. For detailed instructions, see the section called "Inserting ID3 metadata when creating the channel".

Creating a MediaPackage output group

When you planned the workflow for your channel, you might have determined that you want to include a MediaPackage output group. (Or you might have decided to use an HLS output group to deliver to MediaPackage.)

Topics

- Streams section
- Result of this procedure

Procedure to create a MediaPackage output group

Follow these steps to create a MediaPackage output group and its outputs.

To create a MediaPackage output group and its outputs

1. On the **Create channel** page, in the **Output groups** section, choose **Add**. The content pane changes to show the **Add output** group section.
2. Choose **MediaPackage**, and then choose **Confirm**. More sections appear:
   - **MediaPackage destination**
   - **MediaPackage settings**
   - **MediaPackage outputs**–This section shows the single output that is added by default.
3. In the **MediaPackage destination** section, for **MediaPackage channel ID**, enter the channel ID for that channel. For example, `curlinglive`.
4. As part of the planning for this output group, you discussed your requirements with the MediaPackage user. You should have obtained the following information:
   - The ID of the MediaPackage channel. For example, `curlinglive`.
5. (Optional) In the **MediaPackage settings** section, for **Name**, enter a name for the output group.
6. If your plan includes more than one output in this output group, then in **MediaPackage outputs**, choose **Add output** to add the appropriate number of outputs.

   You might want to add an output in order to implement trick-play. For more information about this feature and for instructions on setting it up in the channel, see the section called "Trick-play track via the Image Media Playlist specification".
7. Choose the first **Settings** link to view the sections for the first output. The section contains fields for the output streams (the video, audio, and captions).
8. After you have finished setting up this output group and its outputs, you can create another output group (of any type), if your plan requires it. Otherwise, go to the section called "Save channel".

Streams section

The following fields relate to the encoding of the video, audio, and captions streams (encodes) in the output.

- **Stream settings** section

For information about creating encodes, see the following sections:

- the section called “Set up video”
- the section called “Set up audio”
- the section called “Set up captions”

Packaging of video encodes and audio-only encodes

MediaLive handles the packaging of encodes within each output as follows:

- If an output contains both video and audio (and optionally captions), the audio rendition is marked as **`program audio`**.
- If an output doesn't contain video, the audio rendition is marked as **`audio only`** and each audio encode is marked as **`ALTERNATE_AUDIO_NOT_AUTO_SELECT`**.

Setting the width and height of the video

This section refers to the fields in **Stream settings**, **Video**.

You must specify values in **Width** and **Height**. The MediaPackage output group doesn't support leaving these fields blank to use the width and height from the source video.

Setting the aspect ratio of the video

This section refers to the fields in **Stream settings**, **Video**, **Aspect ratio**.

You must set **PAR control** to **`SPECIFIED`**. The MediaPackage output group doesn't support setting the aspect ratio of the output to follow the source video. When you choose **`SPECIFIED`**, you must complete **PAR numerator** and **PAR denominator**. You can set the **AFD** fields as you want.

Setting the frame rate of the video

This section refers to the fields in **Stream settings**, **Video**, **Frame rate**.

You must set **Framerate control** to **`SPECIFIED`**. The MediaPackage output group doesn't support setting the frame rate of the output to follow the source video. When you choose **`SPECIFIED`**, you must complete **Framerate numerator** and **Framerate denominator**. You can set the scan type as you want; it doesn't relate directly to the frame rate.

Setting up for GOPs and segments

This section refers to the fields in **Stream settings**, **Video**, **GOP structure**.

For the video, you must set the GOP size to ensure that the output from MediaLive has a segment size that is close to the segment size that you specify in MediaPackage. MediaLive and MediaPackage work together to obtain a final segment size. The logic is as follows:

- In MediaLive you specify the **GOP size** and **GOP size units** fields.
- MediaLive calculates the GOP duration, taking into account the frame rate that you specify in the **Video** section of the **Output** page.
- In MediaPackage you specify the segment duration. You always specify a whole number. This segment duration is the *desired* minimum duration.
- When MediaPackage receives the video from MediaLive, it determines how much it must adjust the segment duration to fit a whole number of GOPs into the segment. The segment duration can only be adjusted up, never down. This adjusted segment duration appears in the manifest that MediaPackage produces.

Example 1

Assume that in MediaLive you set the GOP size to 60 frames. You set the frame rate to 29.97. These two values result in a GOP duration of 2.002 seconds.

Assume that in MediaPackage you set the segment duration to 6 seconds. This segment duration is the *desired* minimum duration.

When MediaPackage receives the video from MediaLive, it determines how much it must adjust the segment duration to fit a whole number of GOPs into the segment. In this case, the segment duration must be adjusted to 6.006 seconds (three GOPs, where each GOP is 2.002 seconds long).

Example 2

Assume that in MediaLive, you set the GOP size to 90 frames. You set the frame rate to 30. These two values result in a GOP duration of 3 seconds.

Assume that in MediaPackage you set the segment duration to 4 seconds. This segment duration is the *desired* minimum duration.

When MediaPackage receives the video from MediaLive, it determines how much it must adjust the segment duration to fit a whole number of GOPs into the segment. In this case, the segment duration must be adjusted to 6 seconds (two GOPs, where each GOP is 3 seconds long).

Other encode fields

For information about the fields in each type of encode, see the following sections:

- the section called “Set up video”
- the section called “Set up audio”
- the section called “Set up captions”

Result of this procedure

With a MediaPackage output group, you don't configure as many fields as you do with a regular HLS output group. Instead, MediaLive automatically sets up the output group as follows:

Destination

- The output from pipeline 0 is mapped to the first ingest endpoint in the MediaPackage channel. The output from pipeline 1 (if you have set up a standard channel) is mapped to the second ingest endpoint.

  The mapping of each pipeline to an ingest endpoint never changes. The only change that can occur in the mappings is if you upgrade a single-pipeline input to a standard-class input, or upgrade a single-pipeline channel to a standard channel. In both these cases, pipeline 1 will be mapped to the second ingest endpoint (which has always existed).

  You can view details of the mappings after you have created the channel. Follow the steps in Viewing channel details in the *AWS Elemental MediaPackage User Guide*. In the **Inputs** section, the first item (ingest endpoint) always maps to pipeline 0 in the MediaLive channel, and the second item always maps to pipeline 1.

- The output is delivered to MediaPackage using WebDAV. The output is always a live stream, not a VOD stream.
- The output name or names are automatically set to `Output n`, where n is an integer starting at 1.
- The `nameModifier` for each output is automatically set to match the output name.

Container

- The codec specification is RFC 4281. The player device might use this information.
- The program date time (PDT) period is set to 1 second.
- The PAT interval is set to 0, which means a single PAT is inserted at the beginning of each segment.
- The PMT interval is set to 0, which means a single PMT is inserted at the beginning of each segment.

Resiliency

- Resiliency is handled as follows. If input into MediaLive is lost, then the behavior is for MediaLive to pause delivery. MediaPackage expects this behavior and handles the loss by switching to the other input.

SCTE-35

- Passthrough of SCTE-35 messages is always enabled. If you don't want SCTE-35 markers in the outputs, you can remove them in the channel in AWS Elemental MediaPackage. For information about SCTE-35 handling in a MediaPackage output, see the section called “SCTE-35 message processing”.

ID3

- ID3 metadata is enabled.
- The ability to insert ID3 markers through the output group is disabled. However, you can set up to pass through ID3 markers that are in the input, and you can insert ID3 markers using the MediaLive schedule. For information about ID3 handling in a MediaPackage output, see the section called “ID3 metadata”.

Creating a Microsoft Smooth output group

When you planned the workflow for your channel, you might have determined that you want to include a Microsoft Smooth output group.

Topics

Procedure to create a Microsoft Smooth output group

Follow these steps to create a Microsoft Smooth output group and its outputs.

To create a Microsoft Smooth output group and its outputs

1. On the **Create channel** page, in the **Output groups** section, choose **Add**.
2. In the **Add output group** section, choose **Microsoft Smooth**, and then choose **Confirm**. More sections appear:

   - **Microsoft Smooth group destination** – This section contains fields for the destination of the outputs.
   - **Microsoft Smooth settings** – This section contains fields for the container, the connection to the downstream system, and resiliency.
   - **Microsoft Smooth outputs** – This section shows the single output that is added by default.
   - **Event configuration** – This section contains fields for the destination of the outputs and thecontainer.
   - **Timecode configuration** – This section contains fields for the timecode in the outputs.
   - **Sparse track** – This section contains fields for the container.
3. If your plan includes more than one output in this output group, then in **Microsoft Smooth outputs**, choose **Add output** to add the appropriate number of outputs.
4. In **Microsoft Smooth outputs**, choose the first **Settings** link to view the sections for the first output:

- **Output settings** – This section contains fields for the output destination, and the container.
- **Stream settings** – This section contains fields for the output streams (the video, audio, and captions).

5. (Optional) Enter names for the output group and the outputs:

   - In **Microsoft Smooth settings**, for **Name**, enter a name for the output group. This name is internal to MediaLive; it doesn't appear in the output. For example, **`Sports Curling`**.
   - In the **Output settings** section for each output, for **Output name**, enter a name for the output. This name is internal to MediaLive; it doesn't appear in the output. For example, **`high resolution`**.

6. To complete the other fields, see the topics listed after this procedure.
7. After you have finished setting up this output group and its outputs, you can create another output group (of any type), if your plan requires it. Otherwise, go to the section called "Save channel".

Fields for the output destination

The following fields configure the destination of each Microsoft Smooth output.

- **Output group** – **Microsoft Smooth group destination** section
- **Output group – Event configuration – Event ID mode**
- **Output group – Event configuration – Event ID**
- **Microsoft Smooth settings** section – **General configuration** section:
  - **Connection retry interval**
  - **Num retries**
  - **Filecache duration**
  - **Restart delay**
  - **Certificate mode**

Complete the fields on the console

The full path for each output in a Microsoft Smooth output group consists of the following:

```
URL eventID streamInformation
```

- The URL and event ID are know as the *publishing points*. For example:

  ```
  https://203.0.113.18/sports/Events(1585232182)
  ```

- MediaLive generates the event ID using information that you provide. For more information, expand **Event Configuration** on the console, and choose the **Info** link next to each field.
- MediaLive generates the stream ID. It assigns a unique number to the stream, starting from 0. For example: `/Streams(stream0)`.

  You will be able to see the stream information when you look at the MediaLive logs for the output.

To specify the path and connection to the downstream system

1. When you discussed your requirements with the operator of the Microsoft Smooth downstream system, you should have obtained the following information:

   - The URL for the destination or destinations. For example:

     ```
     https://203.0.113.55/sports/curling
     ```

     ```
     https://203.0.113.82/sports/curling
     ```

   - The user name and password to access the Microsoft IIS server, if the server requires authenticated requests.
   - The settings for connection fields, if the downstream system has special requirements.
2. Complete the **URL** fields in the **Microsoft Smooth group destinations** section. Specify two destinations if the channel is set up as a standard channel, or one destination if it is set up as a single-pipeline channel. Don't worry about the event ID. You will specify that in another field.
3. Complete the **Credentials** section, if the downstream system provided you with a user name and password. For the password, enter the name of the password stored on the AWS Systems Manager Parameter Store. Don't enter the password itself. For more information, see the section called “AWS Systems Manager parameter store”.
4. If you obtained values to configure the connection, enter those values in the **General configuration** section on the **Microsoft Smooth group** page.
5. Set up the event ID in the following fields:

   Output group settings – Event configuration – Event ID Mode

Output group settings – Event configuration – Event ID

You can set up the event ID in three ways:

- With an event ID that you specify – Set **Event ID mode** to **USE_CONFIGURED**. Then specify the ID. For example, **`curling`**. The event ID will look like this: **`/Events(curling)`**
- With a timestamp – Set **Event ID mode** to **USE_TIMESTAMP**. MediaLive generates a Unix timecode based on the time that you start the channel. The event ID will look like this: **`/Events(1585232182)`**
- With no event ID – set **Event ID mode** to **NO_EVENT_ID**. We strongly recommend that you don't use this method.

Fields for the container

The following fields configure the container in each output.

- **Microsoft Smooth settings** section – **General configuration** section – **Fragment length**
- **Event configuration** – **Stream manifest behavior**
- **Event configuration – Event stop behavior**

These fields let you configure some of the streaming behavior. For information about a field, choose the **Info** link in the MediaLive console.

Fields for the encodes

The following fields relate to the encoding of the video, audio, and captions streams (encodes) in the output.

- **Stream settings** section

For information about creating encodes, see the following sections:

- the section called “Set up video”
- the section called “Set up audio”
- the section called “Set up captions”

Fields for other Microsoft Smooth features

Topics

Fields for resiliency

The following field relates to implementing resiliency in a Microsoft Smooth output.

- **Microsoft Smooth output group** – **Microsoft Smooth Settings** section – **General configuration** section – **Input loss action**

Optionally change the value of **Input loss action**.

Choose the **Info** link in the MediaLive console to decide which option to choose.

Fields for timecode

The following fields relate to configuring the timecode and timestamp in all the outputs in the output group.

- **Microsoft Smooth output group** – **Timecode Configuration** section

For details about a field, choose the **Info** link next to the field in the MediaLive console.

Fields for SCTE-35

The following fields relate to configuring the timecode and timestamp in all the outputs in the output group.

- **Microsoft Smooth output group** – **Timecode Configuration** section

If you want all the outputs in this output group to include the SCTE-35 messages that are already present in the input, choose **Sparse track**. The messages will be included in a sparse track. For more information, see the section called “SCTE-35 message processing” and specifically the section called “Enabling decoration – Microsoft Smooth”.

Creating an RTMP output group

When you planned the workflow for your channel, you might have determined that you want to include an RTMP output group.

Topics

Procedure to create an RTMP output group

Follow these steps to create an RTMP output group and its output.

To create an RTMP output group and its output

1. On the **Create channel** page, under **Output groups**, choose **Add**.
2. In the **Add output group** section, choose **RTMP**, and then choose **Confirm**. More sections appear:
   - **RTMP settings** – This section contains fields for the connection configuration, for resiliency, and for captions.
   - **RTMP outputs** – This section shows the single output that is added by default. An RTMP output can contain only one output, so don't click **Add output**.
3. In **RTMP outputs**, choose the **Settings** link to view the sections for the output:
   - **RTMP destination** – This section contains fields for the output destination.
   - **Output settings** – This section contains fields for the connection configuration.
   - **Stream settings** – This section contains fields for the output streams (the video, audio, and captions).
4. (Optional) Enter names for the output group and the output:
   - In **RTMP settings**, for **Name**, enter a name for the output group. This name is internal to MediaLive; it doesn't appear in the output. For example, `Sports Game`.

- In **RTMP output**, in **Output settings**, for **Output name**, enter a name for the output. This name is internal to MediaLive; it doesn't appear in the output.

5. To complete the other fields, see the topics listed after this procedure.
6. After you have finished setting up this output group and its single output, you can create another output group (of any type), if your plan requires it. Otherwise, go to the section called "Save channel".

Fields for the output destination

The following fields configure the location and names of the RTMP output files (the destination).

- **Output** – **RTMP destination** sections

To specify the destination for the output

1. When you discussed your requirements with the operator of the RTMP server, you should have obtained the following information:

   - The protocol for MediaLive to use—RTMP or RTMPS.
   - IP address.
   - Port number.
   - Application name. Also called *app name*.
   - Stream name. Also called *application instance* or *app instance* or *stream key*.

     The operator might give you the application name and stream name as separate pieces of data. Or they might give you a complete path in the format **`string/string`**. In this case, the first string is the application name and the second string is the stream name.
   - The user name and password to access the server, if the downstream system requires authenticated requests.

   Here is an example of the information that the operator will give you:

   `rtmp://203.0.113.17:80/xyz/ywq7b`

   Where `xyz` is the application name, and `ywq7b` is the stream name.
2. Enter the different portions of the destination in the appropriate fields.

Portion of the destination URL	Field
protocol, IP address, port, application name	The two **URL** fields in the **RTMP destinati on** section. (Note that these fields are on the **Output** page, not the **Output group** page.) For example: `rtmp://203.0.113.17:80/xyz` Specify two destinations when the channel is set up as a the section called "Channel class", or one destination when it is set up as a single-pipeline channel.
Stream name	The two **Stream name** fields in the **RTMP outputs** section. For example: **ywq7b**

3. Complete the **Credentials** section, if the server the downstream system provided you with a user name and password. For the password, enter the name of the password stored on the AWS Systems Manager Parameter Store. Don't enter the password itself. For more information, see the section called "AWS Systems Manager parameter store".

Fields for the RTMP connection

The following fields configure the logic for reconnection attempts:

- **RTMP settings – Authentication scheme**
- **RTMP settings – Additional settings – Cache length**
- **RTMP settings – Additional settings – Restart delay**
- **RTMP settings – Additional settings – Cache full behavior**
- **RTMP outputs – Output settings – Connection retry interval**
- **RTMP outputs – Output settings – Num retries**
- **RTMP outputs – Output settings – Additional settings – Certificate mode**

To configure a secure (RTMPS) connection to the destination

1. **Authentication Scheme** – Specify the type of scheme. Typically, choose **`Common`**. Choose **`Akamai`** only if instructed to do so by the downstream system.
2. For **Certificate mode**, choose the option that is required by the downstream system.

   If you connect over RTMP, MediaLive ignores both these fields.

To configure for reconnection

- There are several fields that control how MediaLive behaves if the connection to the RTMP server seems to drop:
  - **Cache length** specifies how long to hold the output in memory, waiting for the RTMP server to respond.
  - When that time expires, **Cache full behavior** specifies whether to disconnect immediately or wait 5 minutes.
  - If MediaLive disconnects, then **Restart delay** specifies how long to wait before trying to reconnect.
  - When MediaLive tries to reconnect, **Connection retry interval** specifies how often to retry. **Num retries** specifies how many times to retry. When the retries expire, this output stops. The channel stops because the single output has lost its connection.

Fields for the video, Audio, and captions streams (encodes)

The following fields relate to the encoding of the video, audio, and captions streams (encodes) in the output.

- **Stream settings** section

For information about creating encodes, see the following sections:

- the section called “Set up video”
- the section called “Set up audio”
- the section called “Set up captions”

Other fields

The following field relates to implementing resiliency in an RTMP output:

- **RTMP settings** – **Input loss action** – For details about a field on the MediaLive console, choose the **Info** link next to the field.

The following field relates to implementing captions in an RTMP output:

- **RTMP settings** – **Caption data** – Complete this field only if at least one of your outputs includes captions with **embedded** as the source captions format and **RTMP CaptionInfo** as the output format. If there are no captions in any output, the value in this field is ignored.

  For detailed information about setting up for captions, see the section called “Captions”.

Creating a UDP output group

When you planned the workflow for your channel, you might have determined that you want to include a UDP output group.

Topics

Procedure to create a UDP output group

Follow these steps to create a UDP output group and its output.

1. On the **Create channel** page, under **Output groups**, choose **Add**.
2. In the **Add output group** section, choose **UDP**, and then choose **Confirm**. More sections appear:

   - **UDP destination** – This section contains fields for the output destination.

- **UDP settings** – This section contains fields for setting up ID3 and for resiliency.
- **UDP outputs** – This section shows the single output that is added by default. A UDP output can contain only one output, so don't click **Add output**.

3. In **UDP outputs**, choose the **Settings** link to view the sections for the output:

   - **Output settings** – This section contains fields for the transport and the connection to the destination.
   - **Stream settings** – This section contains fields for the output streams (the video, audio, and captions).

4. (Optional) Enter names for the output group and the output:

   - In **UDP settings**, for **Name**, enter a name for the output group. This name is internal to MediaLive; it doesn't appear in the output. For example, **`Sports Game`**.
   - In **UDP output**, in **Output settings**, for **Output name**, enter a name for the output. This name is internal to MediaLive; it doesn't appear in the output.

5. To complete the other fields, see the topics listed after this procedure.
6. After you have finished setting up this output group and its single output, you can create another output group (of any type), if your plan requires it. Otherwise, go to the section called "Save channel".

Fields for the output destination

The following fields configure the destination of the output:

- **Output group** – **UDP destination** sections
- **Output** – **Output settings** – **Network settings** – **Buffer msec**

To specify the destination for the output

1. When you discussed your requirements with the operator who manages the downstream system that will receive UDP content, you should have obtained the following information:

   - The URLs
   - The port numbers

For example:

```
udp://203.0.113.28:5000
```

```
udp://203.0.113.33:5005
```

2. Enter the URLs, including the port number, in one or both of the **URL** fields in the **UDP destinations** section.
3. If you enable FEC, leave space between the port numbers for the two destinations.

   For example, if one destination is **`rtp://203.0.113.28:5000`**, assume that FEC also uses port 5002 and 5004. So the lowest possible port number for the other destination is 5005: **`rtp://203.0.113.33:5005`**.
4. (Optional) In the **Output** section, complete the **Buffer msec** field as appropriate. For details, choose the **Info** link next to the field in the MediaLive console.

Fields for the UDP transport

The following fields configure the transport in each output:

- **Output** – **Output settings** – **FEC output settings**, choose a value.
- **Output** – **Output settings** – **Network settings** – **Container settings** section.

Change any values as appropriate. For details about a field, choose the **Info** link next to the field in the MediaLive console.

Fields for the video, audio, and captions stream (encode)

The following fields relate to the encoding of the video, audio, and captions streams (encodes) in the output.

- **Stream settings** section

For information about creating encodes, see the following sections:

- the section called “Set up video”
- the section called “Set up audio”

- the section called “Set up captions”

Fields for other UDP features

The following field relates to implementing resiliency in a UDP output:

- **UDP settings** – **Input loss action** – For details about a field on the MediaLive console, choose the **Info** link next to the field.

The following fields relate to implementing captions in a UDP output:

- **UDP settings** – **Timed metadata ID3 frame type**
- **UDP settings** – **Timed metadata ID3 period**

  Complete these fields if you want to insert timed ID3 metadata or ID3 segment tags into all the outputs in this output group. For detailed instructions, see the section called “ID3 metadata”.

Next steps

After you have set up an output group and its output or outputs, you should create the video, audio, and captions encodes for each output. See *Setup: Creating output encodes*.

Setup: Creating output encodes in an AWS Elemental MediaLive channel

This section describes how to create video, audio, and captions output encodes. You create these encodes as part of the task of creating the output groups and outputs in a channel. You should have already identified and planned the outputs and encodes to include in each output group.

The step of creating encodes is integrated into the step of creating output groups as part of the channel. If you haven't started to created the output groups yet, see *Setup: Creating outputs*.

Topics

Set up the video encode

In the section called “Outputs”, you created the output groups and outputs that you identified when you planned the channel. Each output section contains a **Stream settings** section. You must now create all the video encodes.

General procedure

Follow this general procedure to set up the video encode.

1. Decide how you're going to create each encode:
   - From scratch.
   - By sharing an encode that already exists in this output or another output in the channel.
   - By cloning an encode that already exists in this output or another output in the channel.

   You might have already made this decision. If not, you should decide now. For more information, see the section called “Step 4: Design the encodes”.

   You can share or clone video encodes from one output to another in the same output group, or from one output to an output in another output group.

2. Read the appropriate sections that follow.

Topics

Creating a video encode from scratch in MediaLive

Topics

Setting up the video encodes in most types of output

1. On the **Create channel** page, find the output group that you created.
2. Under that output group, find the output where you want to set up a video encode.
3. Choose the link for the video encode.
4. For **Codec settings**, choose the codec to use for this encode. More fields appear in several sections.
5. Complete each field as appropriate. For details about a field, choose the **Info** link next to the field.

Topics

Width and height (resolution)

For information about the **Width** and **Height** fields (which define the video resolution), choose the **Info** link for each field. The frame rate affects the output charges for this channel. For more information about charges, see the MediaLive price list.

Rate control

For information about the **Rate control** fields, see the section called "Video – rate control mode". There are fields in this section that affect the output charges for this channel. For more information about charges, see the MediaLive price list.

Framerate

For information about the **Framerate** fields, choose the **Info** link for each field. The frame rate affects the output charges for this channel. For more information about charges, see the MediaLive price list.

Codec details

Profile field for H.264

The **Profile** field sets the profile, chroma sampling, and bit depth.

Value in Profile field	Profile	Chroma Sampling	Bit Depth
Baseline	Baseline	4:2:0	8-bit
Main	Main	4:2:0	8-bit
High	High	4:2:0	8-bit
High 10bit	High	4:2:0	10-bit
High 422	High	4:2:2	8-bit
High 422 10bit	High	4:2:2	10-bit

Profile field for H.265

The **Profile** field sets the profile, chroma sampling, and bit depth.

Value in Profile field	Profile	Chroma Sampling	Bit Depth
Main	Main	4:2:0	8-bit
Main_10BIT	Main	4:2:0	10-bit

Timecode

For information about the **Timecode** fields, see the section called "Timecodes".

Color space

For information about the **Color space** fields, see the section called "Video – complex color space conversion".

Additional encoding settings

For information about the **Additional encoding setting**s fields, see the section called "Video – enhanced VQ"

Setting up the video encodes in a Frame Capture output

1. On the **Create channel** page, find the output group that you created.
2. Under that output group, find the output and choose the link for the video encode.
3. Complete each field as appropriate. For details about a field, choose the **Info** link next to the field.
4. When you are ready, go to save the channel.

Sharing a video encode in MediaLive

You can create one video encode and share it among several outputs. Follow the earlier procedure to create the encode once. Then set up the encode for the other outputs using the following steps.

Note that the procedure for sharing a video encode is nearly identical to the procedure for sharing an audio encode or captions encode.

1. On the **Create channel** page, find the output group that you created.

2. Under that output group, find the output where you want to set up a video encode.
3. If the output already contains a video encode, choose that video and then choose **Remove video**.
4. Choose **Add video**. A menu appears that includes the option **Use an existing video description**, followed by a list of the videos that currently exist in the entire channel.
5. Choose the video that you want to use. On the dialog that appears, choose **Share the existing settings**.

   The fields for this encode appear. Above the first field is an information message that lists all the outputs that share this encode.

   You might want to change the video description to include the term *shared*, as a reminder to yourself.

   Keep in mind that there is only one instance of this encode in the channel. Therefore, if you change a field, you will change the field in all the other outputs that use this encode.

   Remember this rule if you change the **Video selector name** field. If you specify a different selector in the encode in one output, you change it in all the outputs that share this encode. If you actually want to specify a different selector, you might need to clone the encode instead of sharing it.

To stop sharing an encode

You might need to stop sharing an encode. For example, you might have outputs A, B, and C that all share the encode H.264-hi-resolution. You want to remove output C from the shared setup and set up output C with its own (unshared) encode.

To stop sharing an encode follow these steps.

1. On the **Create channel** page, find the output group that has the output that contains the video that you want to remove from the shared setup.
2. Select the output group, then select the output that contains the video encode. The name of the shared video encode appears, and the names of all the outputs that share that encode.
3. Make a note of the video encode, in case you need to refer to it again.
4. Select **Remove video**.

You can now create a new video encode for this output, either by creating from scratch, by sharing a different encode, or by cloning the encode that you just unshared (cloning isn't the same as sharing).

Creating a video encode by cloning in MediaLive

You can create one video encode and clone it among several outputs. The *source* encode could be an encode that you created from scratch, or it could be an encode that was itself created by cloning. For example, create *video-1*, then clone it to *video-2*, then clone *video-2* to *video-3*.

Note that the procedure for cloning a video encode is nearly identical to the procedure for cloning an audio encode or captions encode.

1. On the **Create channel** page, find the output group that you created.
2. Under that output group, find the output where you want to set up a video encode.
3. If there is a **Video** button on the left, choose that button and then choose **Remove video**.
4. Choose **Add video**. A menu appears that includes the option **Use an existing video description**, followed by a list of the videos that currently exist in the entire channel.
5. Choose the video encode that you want to use as the source for the new video encode.
6. On the dialog that appears, choose **Clone the existing settings**. The fields for the encode appear, with the fields showing the values from the source encode.
7. Change any fields, as appropriate.

   Keep in mind that this cloned encode is a new encode instance. If you change fields, you don't affect the source encode.

Set up the audio encodes

In the section called “Outputs”, you created the output groups and outputs that you identified when you planned the channel. Each output section contains a **Stream settings** section. You must now create the audio encodes for the outputs.

General procedure

Follow this general procedure to set up the audio encode.

1. Decide how you're going to create each encode:

- From scratch.
- By sharing an encode that already exists in this output or another output in the channel.
- By cloning an encode that already exists in this output or another output in the channel.

You might have already made this decision. If not, you should decide now. For more information, see the section called “Step 4: Design the encodes”.

You can share or clone audio encodes within one output, from one output to another in the same output group, or from one output to an output in another output group.

2. Read the appropriate sections that follow.

Topics

Creating an audio encode from scratch

1. On the **Create channel** page, find the output group that you created.
2. Under that output group, find the output where you want to set up an audio encode.
3. If you need to add a new audio to this output, choose **Add audio**, then choose **Create a new audio description**,
4. Choose the audio encode, and in **Codec settings**, choose the codec to use for this encode. More fields appear.
5. In **Audio selector name**, choose the selector that is the source for this audio encode, according to your plan. You created this selector earlier.
6. Complete other fields as appropriate. For details about a field, choose the **Info** link next to the field.
   - The fields in the **Codec settings** section are different for each type of codec.
   - The fields in the **Remix settings** section are optional.
   - The fields in the **Audio normalization** settings are optional.
   - The fields in the **Additional settings** section are optional.

Creating an audio encode by sharing

You can create one audio encode and share it among several outputs. Follow the earlier procedure to create the encode once. Then set up the encode for the other outputs using the following steps.

Note that the procedure for sharing an audio encode is nearly identical to the procedure for sharing a video encode or captions encode.

1. On the **Create channel** page, find the output group that you created.
2. Under that output group, find the output where you want to set up an audio encode.
3. The output might contain an audio encode that MediaLive has automatically added. If you don't plan to use this audio encode, remove it. Choose the audio encode and choose **Remove audio**.
4. Create a new audio. Choose **Add audio**. A menu appears that includes the option **Use an existing audio description**, followed by a list of the audios that currently exist in the entire channel. Choose the audio that you want to use.
5. On the dialog that appears, choose **Share the existing settings**.

   The fields for this encode appear. Above the first field is an information message that lists all the outputs that share this encode.

   You might want to change the audio description to include the term *shared*, as a reminder to yourself.

   Keep in mind that there is only one instance of this encode in the channel. Therefore, if you change a field, you will change the field in all the other outputs that use this encode.

   Remember this rule if you change the **Audio selector name** field. If you specify a different selector in the encode in one output, you change it in all the outputs that share this encode. If you actually want to specify a different selector, you might need to clone the encode instead of sharing it.

Creating an audio encode by cloning

You can create one audio encode and clone it among several outputs. The *source* encode could be an encode that you created from scratch, or it could be an encode that was itself created by cloning. For example, create *audio-1*, then clone it to *audio-2*, then clone *audio-2* to *audio-3*.

Note that the procedure for cloning an audio encode is nearly identical to the procedure for cloningg a video encode or captions encode.

1. On the **Create channel** page, find the output group that you created.
2. Under that output group, find the output where you want to set up an audio encode.
3. The output might contain an audio encode that MediaLive has automatically added. If you don't plan to use this audio encode, remove it. Choose the audio encode and choose **Remove audio**.
4. Create a new audio. Choose **Add audio**. A menu appears that includes the option **Use an existing audio description**, followed by a list of the audios that currently exist in the entire channel. Choose the audio that you want to use.
5. Choose the audio encode that you want to use as the source for the new audio encode.
6. On the dialog that appears, choose **Clone the existing settings**. The fields for the encode appear, with the fields showing the values from the source encode.
7. Change any fields, as appropriate.

   Keep in mind that this cloned encode is a new encode instance. If you change fields, you don't affect the source encode.

Set up the captions encodes

In the section called “Outputs”, you created the output groups and outputs that you identified when you planned the channel. Each output section contains a **Stream settings** section. You must now create all the captions encodes for the outputs.

General procedure

Follow this general procedure to set up the captions encode.

1. Decide how you're going to create each encode:
   - From scratch.
   - By sharing an encode that already exists in this output or another output in the channel.
   - By cloning an encode that already exists in this output or another output in the channel.

   You might have already made this decision. If not, you should decide now. For more information, see the section called “Step 4: Design the encodes”.

You can share or clone captions encodes within one output, from one output to another in the same output group, or from one output to an output in another output group.

2. Read the appropriate sections that follow.

Topics

Creating a captions encode from scratch

1. On the **Create channel** page, find the output group that you created.
2. Under that output group, find the output where you want to set up a captions encode.
3. If you need to add a new captions to this output, choose **Add captions**, then choose **Create a new captions description**,
4. Choose the captions encode, and in **Codec settings**, choose the format to use for this encode. More fields appear.
5. In **Captions selector name**, choose the selector that is the source for this captions encode, according to your plan. You created this selector earlier.
6. Complete other fields as appropriate, to configure the captions encode. For detailed information about setting up captions encodes, see the section called “Step 4: Set up outputs”.

Creating a captions encode by sharing

You can create one captions encode and share it among several outputs. Follow the earlier procedure to create the encode once. Then set up the encode for the other outputs using the following steps.

Note that the procedure for sharing a captions encode is nearly identical to the procedure for sharing a video encode or captions encode.

1. On the **Create channel** page, find the output group that you created.
2. Under that output group, find the output where you want to set up a captions encode.

3. The output might contain a captions encode that MediaLive has automatically added. If you don't plan to use this captions encode, remove it. Choose the captions encode and choose **Remove captions**.
4. Create a new captions. Choose **Add captions**. A menu appears that includes the option **Use an existing captions description**, followed by a list of the captions that currently exist in the entire channel. Choose the captions that you want to use.
5. On the dialog that appears, choose **Share the existing settings**.

   The fields for this encode appear. Above the first field is an information message that lists all the outputs that share this encode.

   You might want to change the captions description to include the term *shared*, as a reminder to yourself.

   Keep in mind that there is only one instance of this encode in the channel. Therefore, if you change a field, you will change the field in all the other outputs that use this encode.

   Remember this rule if you change the **Captions selector name** field. If you specify a different selector in the encode in one output, you change it in all the outputs that share this encode. If you actually want to specify a different selector, you might need to clone the encode instead of sharing it.

Creating a captions encode by cloning

You can create one captions encode and clone it among several outputs. The *source* encode could be an encode that you created from scratch, or it could be an encode that was itself created by cloning. For example, create *captions-1*, then clone it to *captions-2*, then clone *captions-2* to *captions-3*.

Note that the procedure for cloning a captions encode is nearly identical to the procedure for cloningg a video encode or captions encode.

1. On the **Create channel** page, find the output group that you created.
2. Under that output group, find the output where you want to set up a captions encode.
3. The output might contain a captions encode that MediaLive has automatically added. If you don't plan to use this captions encode, remove it. Choose the captions encode and choose **Remove captions**.

4. Create a new captions. Choose **Add captions**. A menu appears that includes the option **Use an existing captions description**, followed by a list of the captions that currently exist in the entire channel. Choose the captions that you want to use.
5. Choose the captions encode that you want to use as the source for the new captions encode.
6. On the dialog that appears, choose **Clone the existing settings**. The fields for the encode appear, with the fields showing the values from the source encode.
7. Complete other fields as appropriate, to configure the captions encode. For detailed information about setting up captions encodes, see the section called “Step 4: Set up outputs”.
8. Keep in mind that this cloned encode is a new encode instance. If you change fields, you don't affect the source encode.

Next step

After you have created all the output groups, outputs, and output encodes, you are ready to save the channel.

To save (create) the channel, choose **Create channel** in the navigation pane. For more information, see the section called “Save channel”.

Creating an AWS Elemental MediaLive schedule

In AWS Elemental MediaLive, you can manipulate the processing of a channel while it is running. You perform this manipulation by adding actions to the schedule that is associated with the channel. The schedule holds each action until the start time for the action, at which point MediaLive passes the action to the channel, and the channel performs the action.

We recommend that you read this schedule chapter before you start the channel. A key schedule action is input switching, which you must implement if you have a multiple-input channel (if you have attached more than one input to the channel). But there are other actions that might be relevant to your workflow.

Typically, you set up the schedule before you start the channel. At the least, you create schedule actions that you know must occur soon. After you start the channel, you can continue to add schedule actions. You can add actions you already have planned and ad-hoc actions.

Topics

Types of actions in the schedule

The schedule is a list of actions that a channel performs as it is running. You can use actions to do the following:

- Switch the input that the running channel is ingesting.
- Prepare an input that is associated with an immediate input switch, in order to reduce the delay that occurs when MediaLive performs the switch.
- Insert a static image overlay (an image layered over the underlying video) in every output in every output group. This action is called global image overlay.

- Insert a static image overlay into the running channel, only in specific outputs in specific output groups. The action is called per-outputs image overlay.

- Insert a motion graphics overlay into the running channel.
- Insert SCTE 35 messages into the running channel.
- Insert ID3 metadata into the running channel.
- Insert ID3 segment tags into the running channel.
- Pause one or both of the pipelines in the channel.
- Unpause one or both of the pipelines in the channel.

For more information, see the section called “How actions work”.

Types of timing for actions

There are several ways to specify the timing for an action:

- Fixed – Perform the action at a specific time that you specify.

  For most actions, the specified time must be at least 15 seconds in the future. For input prepare actions, the specified time must be at least 15 seconds before the start of the associated input switch.
- Immediate – Perform the action as soon as possible.

  You don't specify a time.
- Follow – Perform the action just before the specified input switch starts, or just after the currently running input has finished.

The following table shows the types of timing that apply to each type of action. To read this table, find an action in the first column, then read across the row for the applicable types of timing.

Type of action	Supported types of timing		
	Fixed	**Follow (Note A)**	**Immediate**
Switch the input (perform an input switch)	Yes	Yes	Yes
Prepare the input (perform an input prepare)	Yes	Yes	Yes
Activate a global static image overlay	Yes		Yes
Activate a per-outputs static image overlay	Yes		Yes
Activate a motion graphics overlay	Yes		Yes
Deactivate a global static image overlay	Yes		Yes
Deactivate a per-outputs static image overlay	Yes		Yes
Deactivate a motion graphics overlay	Yes		Yes
Insert a SCTE 35 message	Yes	Yes	Yes
Insert ID3 metadata	Yes		Yes
Insert an ID3 segment tag	Yes		Yes

Type of action	Supported types of timing		
	Fixed	Follow (Note A)	Immediate
Pause or unpause one or both pipelines	Yes		Yes

Note A

With a follow, the applicable action can follow an input switch. It can't follow other types of actions. Therefore, the action that is *being followed* is always an input switch. The action that does the follow is an input switch, an input prepare, or a SCTE 35 message.

How schedule actions work

This section describes how MediaLive handles each combination of action type and start type.

Topics

How input switch actions work

You can set up an action to switch the input that the running channel is ingesting. The channel stops ingesting the current input and starts ingesting the specified input.

The input must already be attached to the channel.

Before you add input switching actions to the schedule, read the section called “Input switching”.

Input switch with fixed start

When you create the action, you include a start time. The start time for the action must be at least 15 seconds in the future but not more than 14 days in the future. After that cutoff, MediaLive rejects the request to create the action.

After you have created the action, the action sits in the schedule. Approximately 15 seconds before the start time, the schedule passes the action to the channel. The channel sets up so that the input switches at the specified time.

Input switch with immediate start

When you create the action, you set the start type to *immediate*.

For an input switch in a standard channel (a channel with two pipelines), MediaLive internally sets the start time to 10 seconds in the future. This delay ensures that the switch occurs at exactly the same time for the two pipelines.

The schedule immediately passes the action to the channel. The channel immediately starts to switch the input (for a single-pipeline channel), or sets up to switch at the specified time (for a standard channel).

Input switch with follow start

When you create the action, you specify the input switch action that you want this action to follow. That *reference action* must be an input switch.

The input for the reference action must have a source end behavior of *Continue*. To find the **Source end behavior** field, go to the **Create channel** page, find the input in the **Input attachment** list, and then find **General input settings**.

After you create the action, the action waits in the schedule. Just before the reference action is due to finish, the schedule passes the action to the channel so that the channel can switch to the new input as soon as the current input has finished.

How input prepare actions work

You can set up an action to prepare an input that is associated with an immediate input switch, in order to reduce the delay that occurs when MediaLive performs the switch.

The input must already be attached to the channel. However, there is no requirement for the input switch for this input to already exist in the schedule. For example, input X must be *attached* to the

channel. You can create action A to prepare input X and later on you can create action B to switch to input X. Or you can create action B and then create action A.

Before you add input prepare actions to the schedule, read the section called “Input prepare”.

Input prepare with fixed start

When you create the action, include a start time. The start time for the action must be at least 15 seconds before the start time of the associated input switch, but not more than 14 days in the future. After that cutoff, MediaLive rejects the request to create the action.

After you have created the action, the action sits in the schedule. Approximately 15 seconds before the start time of the prepare action, the schedule passes the action to the channel. The channel starts preparing the input.

Input prepare with immediate start

When you create the action, you set the start type to *immediate*.

The schedule immediately passes the action to the channel. The channel immediately starts the prepare.

Input prepare with follow start

When you create the action, you specify the input switch action that you want this action to follow. That *reference action* must be an input switch.

The input for the reference action must have a source end behavior of *Continue*. To find the **Source end behavior** field, go to the **Create channel** page, find the input in the **Input attachment** list, and then find **General input settings**.

After you create the action, the action waits in the schedule. Just before the reference action is due to finish, the schedule passes the action to the channel. As soon as the current input has finished, the channel switches to the new input.

How image overlay actions work

You can set up an action to insert and remove an image overlay on the video:

- The activate action inserts an image overlay and activates it so that it is superimposed on the underlying video. If the image overlay information includes a duration, then at the appropriate time the image overlay is removed.

- The deactivate action removes an image overlay. You therefore use this action to remove a currently running image overlay before the specified duration, or remove it when no duration is specified.

Before you add image overlay actions to the schedule, read the section called “Image overlays”.

Global or per-outputs insertion

There are two ways to insert image overlays:

- Use the **Static image activate** feature to insert globally: You can create an action to insert an image overlay in every output in every output group in a channel.

- Use the **Static image output activate** feature to insert per-output: You can create an action to insert an image overlay in specific outputs in specific output groups in a channel.

Activate or deactivate with fixed start

When you create the action, you include a start time. The start time for the action must be at least 15 seconds in the future but not more than 14 days in the future. After that cutoff, MediaLive rejects the request to create the action.

After you have created the action, the action sits in the schedule. Approximately 15 seconds before the start time, the schedule passes the action to the channel. At the start time, the channel inserts the image overlay or removes the image overlay from the video.

Activate or deactivate with immediate start

When you create the action, you set the start type to *immediate*.

The schedule immediately passes the action to the channel. The channel immediately inserts the image overlay or removes the image overlay.

How motion graphics overlay works

You can set up an action to insert and remove a motion graphics overlay on the video:

- The activate motion graphics action inserts a motion graphic and activates it so that it is superimposed on the underlying video. If the image overlay information includes a duration, then at the appropriate time the motion graphic is removed.

- The deactivate motion graphics action removes an image overlay. You therefore use this action to remove a currently running motion graphics before the specified duration, or remove it when no duration is specified.

For information about preparing the motion graphics asset that the action inserts, see the section called "Motion graphics overlay".

Activate or deactivate with fixed start

When you create the action, you include a start time. The start time for the action must be at least 15 seconds in the future but not more than 14 days in the future. After that cutoff, MediaLive rejects the request to create the action.

After you have created the action, the action sits in the schedule. Approximately 15 seconds before the start time, the schedule passes the action to the channel. At the start time, the channel inserts the motion graphic or removes the motion graphic from the video.

Activate or deactivate with immediate start

When you create the action, you set the start type to *immediate*.

The schedule immediately passes the action to the channel. The channel immediately inserts the motion graphic or removes the motion graphic.

How SCTE 35 actions work

You can set up an action to insert a SCTE 35 message in the channel. There are three types of actions:

- Action to insert a splice_insert into the channel: a SCTE 35 message with splice_command_type set to splice_insert.
- Action to insert a time_signal into the channel: a SCTE 35 message with splice_command_type set to time_signal.
- Action to insert a SCTE 35 return-to-network message into the schedule in order to end a splice_insert that either has a duration or has no duration.

Before you add SCTE 35 actions to the schedule, read the section called "SCTE-35 message processing".

Insert SCTE 35 message with fixed start

When you create the action, you include a start time. The start time for the action must be at least 15 seconds in the future but not more than 14 days in the future. After that cutoff, MediaLive rejects the request to create the action.

After you have created the action, the action sits in the schedule. Approximately 15 seconds before the start time, the schedule passes the action to the channel. At the start time, the channel inserts the SCTE 35 message into the stream.

After the channel inserts the message, MediaLive processes the inserted message in the same way as it processes messages that were already in the source content.

Insert SCTE 35 message with immediate start

When you create the action, you set the start type to *immediate*.

The schedule immediately passes the action to the channel. The channel immediately inserts the SCTE 35 message into the stream.

After the channel inserts the message, MediaLive processes the inserted message in the same way as it processes messages that were already in the source content.

Insert SCTE 35 message with follow start

When you create the action, you specify the input switch action that you want this action to follow. That *reference action* must be an input switch.

The input for the reference action must have a source end behavior of *Continue*. To find the **Source end behavior** field, go to the **Create channel** page, find the input in the **Input attachment** list, and then find **General input settings**.

After you create the action, the action waits in the schedule. Just before the reference action is due to finish, the schedule passes the action to the channel. As soon as the current input has finished, the channel inserts the SCTE 35 message into the stream.

After the channel inserts the message, MediaLive processes the inserted message in the same way as it processes messages that were already in the source content.

How ID3 metadata actions work

You can set up an action to insert ID3 data in the channel. The action inserts ID3 metadata in outputs where ID3 passthrough is enabled. You must specify a fully formed ID3 metadata item

(including both a header and a frame, as per the ID3 specification) and encode it as base64. MediaLive inserts the metadata once, at the time that you specify.

Before you add ID3 metadata actions to the schedule, read the section called "ID3 metadata".

Insert ID3 metadata with fixed start

When you create the action, you include a start time. The start time for the action must be at least 15 seconds in the future but not more than 14 days in the future. After that cutoff, MediaLive rejects the request to create the action.

After you have created the action, the action sits in the schedule. Approximately 15 seconds before the start time, the schedule passes the action to the channel. At the start time, the channel inserts the data into the channel.

Insert ID3 metadata with immediate start

When you create the action, you set the start type to *immediate*.

The schedule immediately passes the action to the channel. The channel immediately inserts the data into the channel.

How ID3 segment tag actions work

You can set up an action to insert ID3 data in each segment in HLS and MediaPackage output packages where ID3 passthrough is enabled. There are two types of segment tags:

- Tag: You specify the `value` for a `TXXX` field inside the ID3 tag. MediaLive creates an ID3 tag with a single TXXX field and inserts the tag in every segment.

- ID3: You specify a fully formed ID3 metadata item (including both a header and a frame, as per the ID3 specification) and encode it as base64. MediaLive inserts the tag in every segment.

Before you add ID3 segment tag actions to the schedule, read the section called "ID3 segment tags".

Insert ID3 segment tags with fixed start

When you create the action, you include a start time. The start time for the action must be at least 15 seconds in the future but not more than 14 days in the future. After that cutoff, MediaLive rejects the request to create the action.

After you have created the action, the action sits in the schedule. Approximately 15 seconds before the start time, the schedule passes the action to the channel. At the start time, the channel inserts the data into the channel.

Insert ID3 data with immediate start

When you create the action, you set the start type to *immediate*.

The schedule immediately passes the action to the channel. The channel immediately inserts the data into the channel.

How pause and unpause actions work

You can insert an action to pause and unpause one or both pipelines in the channel. The action pauses the specified pipelines and unpauses any unspecified pipelines:

- Action with *one* pipeline specified–The action pauses the specified pipeline and unpauses the other pipeline.
- Action with *both* pipelines specified–The action pauses both pipelines.
- Action with *no* pipelines specified–The action unpauses both pipelines.

> **Note**
>
> The pipelines that you don't specify are not left in their current state. They are always set to unpaused.

Pause or unpause with fixed start

When you create the action, you include a start time. The start time for the action must be at least 15 seconds in the future but not more than 14 days in the future. After that cutoff, MediaLive rejects the request to create the action.

After you have created the action, the action sits in the schedule. Approximately 15 seconds before the start time, the schedule passes the action to the channel. At the start time, the channel pauses or unpauses the pipelines in the channel.

Pause or unpause with immediate start

When you create the action, you set the start type to *immediate*.

The schedule immediately passes the action to the channel. The channel immediately pauses or unpauses the pipelines in the channel.

Working with the schedule (console)

You can use the AWS Elemental MediaLive console to create or delete any of the schedule actions in a channel. You can work with the schedule when the channel is running or when it is idle.

The actions are performed in the channel when the channel is running.

The console provides two views for working with actions:

- A list view that lists actions in tabular format
- A timeline view that shows a timeline representation of the actions

In either view, you can do the following:

- Create individual actions.
- Delete individual actions.
- Delete several actions in one request (a batch command).
- View the actions currently in the schedule.

Topics

Creating actions in the schedule (console)

You can create different actions in the schedule. For a list of supported actions, see the section called “Types of actions”.

The general procedure is the same to create any type of action.

To create an action

1. Read the information about planning the actions you want to add.
2. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
3. In the navigation pane, choose **Channel**, and then choose the channel that you want to work with.
4. On the **Details** pane, choose the **Schedule** tab.
5. Choose the **Switch** button to display the view that you want: **List** view or **Timeline** view. For information about the layout and color coding of the timeline view, see the section called "Viewing the schedule".
6. For **List** view, choose the appropriate action:

   - To create a fixed, follow, or immediate action from scratch, choose **Create**.
   - To create a follow action after an existing action, choose that action, choose **Schedule actions**, and then choose **Create follow actions from**.

     This method displays the **Create schedule action** page with some fields already completed, so you can quickly create a follow switch for that existing action.
7. For **Timeline** view, choose the appropriate action:

   - To create a fixed, follow, or immediate action from scratch, choose **Create**.
   - To create a follow action, find the input switch that you want to follow, and then choose **Create follow action** in that card.

     This method displays the **Create schedule action** page with some fields already completed, so you can quickly create a follow switch for that existing action.
8. On the **Create schedule action** page, complete the fields. For information about completing the fields, see the following topics.
9. When you have finished, choose **Create**.

   MediaLive adds the action to the list or the timeline at its appropriate time slot.

When you create a follow input switch, you effectively create an *input follow chain*. The input follow chain starts with the input above the first follow and ends with the last follow input. For

more information about input follow chains, see the section called "Fixed, immediate, and follow switches".

Topics

Fields for an input switch

This section describes how to complete the fields for these three types of input switches:

- A switch to a static live input
- A switch to a static file input
- A switch to a dynamic file input

Topics

- Fields for a switch to a dynamic file input

Fields for a switch to a static live input

This table shows the fields that apply for an action to switch to a static live input.

Field	Description
Action name	A name for this input switch.
Action type	Input Switch.
Input attachment	The live input to switch to. The input must already be set up as an input attachment in this channel. For example, if you want to switch from input A to input B, specify input B in this field.
Start type	**Fixed**, **Immediate**, or **Follow**. For information about start types for input prepare, see the section called "Fixed, immediate, and follow switches".
Date and time	If the **Start type** is **Fixed**, specify the date and time (in UTC format) that the channel must switch to this new input. This time must be at least 30 seconds in the future. Note that the time is the wall clock time, not the timecode in the input.
Reference action name	If the **Start type** is **Follow**, choose the input to switch from, which is the input that precedes this new input. The dropdown list shows all existing input switches that are file inputs. Remember that input B can follow input A only if input A is a file input and the source end behavior for input A is *continue*.

Field	Description
	For information about these switching rules, see the section called "Fixed, immediate, and follow switches". For example, if you want to switch from input A to input B, specify input A in this field.
Follow point	If the **Start type** is **Follow**, complete this field. The follow point is always **End**, to indicate that the switch will occur when the input in **Reference action name** has finished.

Fields for a switch to a static file input

This table shows the fields that apply for an action to switch to a static file input.

Field	Description
Action name	A name for this input switch.
Action type	Input Switch.
Input attachment	The file input to switch to. The input must already be set up as an input attachment in this channel. For example, if you want to switch from input A to input B, specify input B in this field.
Input clippings settings – Enable input clipping	This field appears only for a file input that is eligible for input clipping. Enable the field if you want to clip the file at the start and end, or only at the start, or only at the end.

Field	Description
Input clippings settings – **Input timecode source**	Choose the source: Zero-based – To set the start and end times relative to the start of the file, which is 00:00:00:00. Embedded – To set the times based on the timecode in the file. The file must have a timecode, otherwise the clipping instruction is ignored.
Input clippings settings – **Start timecode**, **Stop timecode**	Complete one or both fields. Enter values in the format hh:mm:ss:ff.
Input clippings settings – **Last frame clipping behavior**	This field appears only if you specify a stop timecode. Exclude last frame – Clip the file before the frame specified in the end timecode. For example, if the end timecode is 01:00:30:19, don't include frame 19. Include last frame – Don't clip the file. In the preceding example, include frame 19.
Start type	**Fixed**, **Immediate**, or **Follow**. For information about start types for input prepare, see the section called “Fixed, immediate, and follow switches”.
Date and time	If the **Start type** is **Fixed**, specify the date and time (in UTC format) that the channel must switch to this new input. This time must be at least 30 seconds in the future. Note that the time is the wall clock time, not the timecode in the input.

Field	Description
Reference action name	If the **Start type** is **Follow**, choose the input to switch from, which is the input that precedes this new input. The dropdown list shows all existing input switches that are file inputs. Remember that input B can follow input A only if input A is a file input. For information about these switching rules, see the section called "Fixed, immediate, and follow switches". For example, if you want to switch from input A to input B, specify input A in this field.
Follow point	If the **Start type** is **Follow**, complete this field. The follow point is always **End**, to indicate that the switch will occur when the input in **Reference action name** has finished.

Fields for a switch to a dynamic file input

This table shows the fields that apply for an action to switch to a dynamic file input.

Field	Description
Action name	A name for this input switch.
Action type	Input Switch.
Input attachment	The file input to switch to. The input must already be set up as an input attachment in this channel. For example, if you want to switch from input A to input B, specify input B in this field.
Dynamic input setting – URL path for input source A	This field appears if the input is set up as a dynamic input. Enter a value to replace the

Field	Description
	`$urlPath$` portion of the URL for source A in the input. A hint below the fields shows the URL path that you created for this source.
Dynamic input setting – **Use the same URL path for input source B**	This field appears if the input is attached to a standard channel, meaning that it has two pipelines and therefore has two sources. Enable this field (the default) if you want to use the same value for the `$urlPath$` in source A and source B. Disable this field to use a different value, then enter the value.
Input clippings settings – **Enable input clipping**	This field appears only for a file input. Enable the field if you want to clip the file at the start and end, or only at the start, or only at the end.
Input clippings settings – **Input timecode source**	Choose the source: Zero-based – To set the start and end times relative to the start of the file, which is 00:00:00:00. Embedded – To set the times based on the timecode in the file. The file must have a timecode, otherwise the clipping instruction is ignored.
Input clippings settings – **Start timecode**, **Stop timecode**	Complete one or both fields. Enter values in the format hh:mm:ss:ff.

Field	Description
Input clippings settings – Last frame clipping behavior	This field appears only if you specify a stop timecode. Exclude last frame – Clip the file before the frame specified in the end timecode. For example, if the end timecode is 01:00:30:19, don't include frame 19. Include last frame – Don't clip the file. In the preceding example, include frame 19.
Start type	**Fixed**, **Immediate**, or **Follow**. For information about start types for input prepare, see the section called "Fixed, immediate, and follow switches".
Date and time	If the **Start type** is **Fixed**, specify the date and time (in UTC format) when the channel must switch to this new input. This time must be at least 30 seconds in the future. Note that the time is the wall clock time, not the timecode in the input.
Reference action name	If the **Start type** is **Follow**, choose the input to switch from, which is the input that precedes this new input. The dropdown list shows all existing input switches that are file inputs. Remember that input B can follow input A only if input A is a file input. For information about these switching rules, see the section called "Fixed, immediate, and follow switches". For example, if you want to switch from input A to input B, specify input A in this field.

Field	Description
Follow point	If the **Start type** is **Follow**, complete this field. The follow point is always **End**, to indicate that the switch will occur when the input in **Reference action name** has finished.

Fields for an input prepare

This section describes how to complete the fields for these three types of input prepares:

- A prepare of a static live input
- A prepare of a static file input
- A prepare of a dynamic file input

Fields for a prepare of a static live input

This table shows the fields that apply for an action to prepare a static live input.

Field	Description
Action name	A name for this input prepare.
Action type	Input Prepare.
Input attachment	The live input to prepare. The input must already be set up as an input attachment in this channel.
Start type	**Fixed**, **Immediate**, or **Follow**. For information about start types for input prepare, see the section called "Types of starts".
Date and time	If the **Start type** is **Fixed**, specify the date and time (in UTC format) that the channel must start to prepare the input. This time should be

Field	Description
	at least 10 seconds before the upcoming input *switch*. Note that the time is the wall clock time, not the timecode in the input.
Reference action name	If the **Start type** is **Follow**, choose the input to follow. This input is the input whose end you want to use as the trigger for the input prepare. It is not the input for the upcoming input switch. The dropdown list shows all existing input switches. If the input switch that you want to use as the reference (trigger) isn't listed, you need to first create that input switch. For information about start types for input prepare, see the section called “Types of starts”.
Follow point	If the **Start type** is **Follow**, complete this field. The follow point is always **End**, to indicate that the input prepare will occur when the input in **Reference action name** has finished.

Fields for a prepare of a static file input

This table shows the fields that apply for an action to prepare a static file input.

Field	Description
Action name	A name for this input prepare.
Action type	Input Prepare.

Field	Description
Input attachment	The file input to prepare. The input must already be set up as an input attachment in this channel.
Input clippings settings – Enable input clipping	This field appears only for a file input that is eligible for input clipping. Enable the field if the upcoming input switch (which you are preparing) will also be clipped. You must set up the prepare action with identical clipping instructions as the switch action.
Input clippings settings – Input timecode source	Choose the source: Zero-based – To set the start and end times relative to the start of the file, which is 00:00:00:00. Embedded – To set the times based on the timecode in the file. The file must have a timecode, otherwise the clipping instruction is ignored. Make sure you choose the same source in this prepare action and the upcoming switch action.
Input clippings settings – Start timecode, **Stop timecode**	Complete one or both fields. Enter values in the format hh:mm:ss:ff. Make sure you enter the same values in this prepare action and the upcoming switch action.

Field	Description
Input clippings settings – Last frame clipping behavior	This field appears only if you specify a stop timecode. Exclude last frame – Clip the file before the frame specified in the end timecode. For example, if the end timecode is 01:00:30:19, don't include frame 19. Include last frame – Don't clip the file. In the preceding example, include frame 19. Make sure you choose the same option in this prepare action and the upcoming switch action.
Start type	**Fixed**, **Immediate**, or **Follow**. For information about start types for input prepare, see the section called "Types of starts".
Date and time	If the **Start type** is **Fixed**, specify the date and time (in UTC format) that the channel must start to prepare this input. This time should be at least 10 seconds before the upcoming input *switch*. Note that the time is the wall clock time, not the timecode in the input.

Field	Description
Reference action name	If the **Start type** is **Follow**, choose the input to follow. This input is the input whose end you want to use as the trigger for the input prepare. It is not the input for the upcoming input switch. The dropdown list shows all existing input switches. If the input switch that you want to use as the reference (trigger) isn't listed, you need to first create that input switch. For information about start types for input prepare, see the section called "Types of starts".
Follow point	If the **Start type** is **Follow**, complete this field. The follow point is always **End**, to indicate that the input prepare will occur when the input in **Reference action name** has finished.

Fields for a prepare of a dynamic file input

This table shows the fields that apply for an action to prepare a dynamic file input.

Field	Description
Action name	A name for this input prepare.
Action type	Input Prepare.
Input attachment	The file input to prepare. The input must already be set up as an input attachment in this channel.
Dynamic input setting – URL path for input source A	This field appears if the input is set up as a dynamic input.

Field	Description
	Enter a value to replace the `$urlPath$` portion of the URL for source A in the input. A hint below the fields shows the URL path that you created for this source. Make sure you enter the same values in this prepare action and the upcoming switch action.
Dynamic input setting – Use the same URL path for input source B	This field appears if the input is attached to a standard channel, meaning that it has two pipelines and therefore has two sources. Enable this field (the default) if you want to use the same value for the `$urlPath$` in source A and source B. Disable this field to use a different value, then enter the value. Make sure you choose the same option in this prepare action and the upcoming switch action.
Input clippings settings – Enable input clipping	This field appears only for a file input that is eligible for input clipping. Enable the field if the upcoming input switch (which you are preparing) will also be clipped. You must set up the prepare action with identical clipping instructions as the switch action.

Field	Description
Input clippings settings – **Input timecode source**	Choose the source: Zero-based – To set the start and end times relative to the start of the file, which is 00:00:00:00. Embedded – To set the times based on the timecode in the file. The file must have a timecode, otherwise the clipping instruction is ignored. Make sure you choose the same source in this prepare action and the upcoming switch action.
Input clippings settings – **Start timecode**, **Stop timecode**	Complete one or both fields. Enter values in the format hh:mm:ss:ff. Make sure you enter the same values in this prepare action and the upcoming switch action.
Input clippings settings – **Last frame clipping behavior**	This field appears only if you specify a stop timecode. Exclude last frame – Clip the file before the frame specified in the end timecode. For example, if the end timecode is 01:00:30:19, don't include frame 19. Include last frame – Don't clip the file. In the preceding example, include frame 19. Make sure you choose the same option in this prepare action and the upcoming switch action.

Field	Description
Start type	**Fixed**, **Immediate**, or **Follow**. For information about start types for input prepare, see the section called "Types of starts".
Date and time	If the **Start type** is **Fixed**, specify the date and time (in UTC format) that the channel must start to prepare this input. This time should be at least 10 seconds before the upcoming input *switch*. Note that the time is the wall clock time, not the timecode in the input.
Reference action name	If the **Start type** is **Follow**, choose the input to follow. This input is the input whose end you want to use as the trigger for the input prepare. It is not the input for the upcoming input switch. The dropdown list shows all existing input switches. If the input switch that you want to use as the reference (trigger) isn't listed, you need to first create that input switch.
Follow point	If the **Start type** is **Follow**, complete this field. The follow point is always **End**, to indicate that the input prepare will occur when the input in **Reference action name** has finished.

Fields for activating a global image overlay

This table shows the fields that apply for an action to activate an image overlay.

Field	Description
Action type	Static Image Activate.
Action name	A name for this activation action. For example, the layer and the name of the image to overlay.
Start type	**Fixed** or **Immediate**.
Date and time	The date and time (in UTC format) that the channel must activate the image overlay. The time should be at least 60 seconds later than the time that you submit the action. Note that the time is the wall clock time, not the timecode in the input.
Input location	Enter the locations (URLs) on the server where the image file is stored. Also complete **Credentials**, if the server requires that you provide user credentials.
Other fields	Complete these fields to control the layer, position, look (such as fade-in), and other behavior of the image.

Fields for deactivating a global image overlay

This table shows the fields that apply for an action to deactivate an image overlay.

Field	Description
Action type	Static Image Deactivate.
Action name	A name for this deactivation action. For example, the name of the image. Or a name

Field	Description
	that ties back to the activation action plus the term "deactivate."
Start type	**Fixed** or **Immediate**.
Date and time	If the **Start type** is **Fixed**, specify the date and time (in UTC format) that the channel must deactivate the image overlay. The time should be at least 60 seconds later than the time that you submit the action. Note that the time is the wall clock time, not the timecode in the input.
Layer	Enter the layer that contains the image overlay that you want to deactivate. A value 0 to 7. Default is 0.
Fade out	Enter the time in milliseconds for the image to fade out. Default is 0 (no fade-out).

Fields for activating a per-outputs image overlay

This table shows the fields that apply for an action to activate an image overlay.

Field	Description
Action type	Static Image Output Activate.
Action name	A name for this activation action. For example, the layer and the name of the image to overlay.
Start type	**Fixed** or **Immediate**.
Date and time	The date and time (in UTC format) that the channel must activate the image overlay. The

Field	Description
	time should be at least 60 seconds later than the time that you submit the action. Note that the time is the wall clock time, not the timecode in the input.
Input location	Enter the locations (URLs) on the server where the image file is stored. Also complete **Credentials**, if the server requires that you provide user credentials.
Other fields	Complete these fields to control the layer, position, look (such as fade-in), and other behavior of the image.

Fields for deactivating a per-outputs image overlay

This table shows the fields that apply for an action to deactivate the image. The deactivate action operates on one per-output layer that you specify and on the outputs that you specify. It doesn't operate on a specific image.

To understand how this statement is significant, consider this example. You might specify layer 4 and outputs A and C. Layer 4 in output A contains the image overlay X, output B contains image overlay X, and output C contains image X. MediaLive removes X from output A, and removes Y from output C. Image X in output B will remain active.

Field	Description
Action type	Static Image Output Deactivate.
Action name	A name for this deactivation action. For example, the name of the image. Or a name that ties back to the activation action plus the term "deactivate."
Start type	**Fixed** or **Immediate**.

Field	Description
Date and time	If the **Start type** is **Fixed**, specify the date and time (in UTC format) that the channel must deactivate the image overlay. The time should be at least 60 seconds later than the time that you submit the action. Note that the time is the wall clock time, not the timecode in the input.
Outputs	Select the output or outputs where you want to deactivate a specific layer.
Layer	Identify the layer that you want to deactivate. Enter a value 0 to 7. Default is 0.
Fade out	Enter the time in milliseconds for the image to fade out. Default is 0 (no fade-out).

Fields for activating a motion graphics overlay

This table shows the fields that apply for an action to activate a motion graphics overlay.

Field	Description
Action type	Motion Graphics Activate.
Action name	A name for this activation action. For example, the name of the motion graphic asset.
Start type	**Fixed** or **Immediate**.
Date and time	The date and time (in UTC format) that the channel must activate the motion graphics overlay. The time should be at least 60 seconds later than the time that you submit the action.

Field	Description
	Note that the time is the wall clock time, not the timecode in the input.
Duration	Optional. The duration in milliseconds for the motion graphic to remain on the video. If you omit this field or set it to 0, the duration is unlimited and the motion graphic will remain until you create a deactivate action.

URL	The URL of the motion graphics asset. This asset is always an HTML file. The URL follows this syntax:
	
	`<protocol>://<path>/<file>.html`
	
	For example:
	
	`https://example.com/ticker_tape.html`

Credentials	Complete this section only if the server where the motion graphics asset is stored requires user authentication from MediaLive. Enter the user name provided by the owner of the server. For the password, enter the name of the password stored on the AWS Systems Manager Parameter Store. Don't enter the password itself. For more information, see the section called “AWS Systems Manager parameter store”.

Fields for deactivating a motion graphics overlay

This table shows the fields that apply for an action to deactivate an motion graphics overlay.

Field	Description
Action type	Motion Graphics Deactivate.
Action name	A name for this deactivation action. For example, **`deactivate_motion_graphic`** .
Start type	**Fixed** or **Immediate**.
Date and time	If the **Start type** is **Fixed**, specify the date and time (in UTC format) that the channel must deactivate the motion graphics overlay. The time should be at least 60 seconds later than the time that you submit the action. Note that the time is the wall clock time, not the timecode in the input.

Fields for a splice_insert message

This table shows the fields that apply for an action to insert a splice_insert SCTE 35 message.

Field	Description
Action type	SCTE 35 Splice Insert.
Action name	A name for this splice_insert action. For example, splice_insert actions could be numbered sequentially, restarting every day or every month.
Start type	**Fixed** or **Follow** or **Immediate**.
Date and time	If the **Start type** is **Fixed**, specify the UTC start time for the splice_insert action. The time should be at least 15 seconds in the future.

Field	Description
	Note that the time is the wall clock time, not the timecode in the input.
Reference action name	If the **Start type** is **Follow**, choose the input to follow. The dropdown list shows all existing input switches that are file inputs. Remember that a SCTE 35 action can follow input A only if input A is a file input and the source end behavior for input A is *continue*. For information about these switching rules, see the section called “Fixed, immediate, and follow switches”.
Follow point	If the **Start type** is **Follow**, complete this field. The follow point is always **End**, to indicate that the switch will occur when the input in **Reference action name** has finished.
Splice event id	The ID for the splice event. Enter an ID for the splice event that is unique among all scheduled and active splice_insert messages in this channel. A message is active if the schedule action is in process in the channel and has not completed.
Duration	The duration for the splice event. Complete in one of these ways: • Enter the duration, in 90-kHz ticks. For example, 1350000, which is equal to 15 seconds. • Leave empty to create a message with no duration.

The splice_insert inserted in the transport stream will have the following:

```
segmentation_event_cancel_indicator = 0
out_of_network = 1
duration_flag = 1
duration = the specified time
```

Or

```
segmentation_event_cancel_indicator = 0
out_of_network = 1
duration_flag = 0
```

Fields for a time_signal message

This table shows the fields that apply for an action to insert a time_signal SCTE 35 message.

Field	Description
Action type	SCTE 35 Time Signal.
Action name	A name for this time_signal action. For example, you might name time_signal actions with a sequential number, restarting every day or every month.
Start type	**Fixed** or **Follow** or **Immediate**.
Date and time	If the **Start type** is **Fixed**, specify the UTC start time for the time_signal. The time should be at least 15 seconds in the future. Note that the time is the wall clock time, not the timecode in the input.
Reference action name	If the **Start type** is **Follow**, choose the input to follow. The dropdown list shows all existing input switches that are file inputs. Remember

Field	Description
	that a SCTE 35 action can follow input A only if input A is a file input and the source end behavior for input A is *continue*. For information about these switching rules, see the section called "Fixed, immediate, and follow switches".
Follow point	If the **Start type** is **Follow**, complete this field. The follow point is always **End**, to indicate that the switch will occur when the input in **Reference action name** has finished.
Add Scte35 descriptors	Choose this button and complete the fields that appear. The descriptors are a standard component of a time_signal message.

Fields for a return-to-network message

This table shows the fields that apply for an action to insert a return-to-network SCTE 35 message.

Field	Description
Action type	SCTE 35 Return to Network.
Action name	A name for this return-to-network action. For example, **`splice0003_ return_early`** .
Start type	**Fixed** or **Follow** or **Immediate**.
Date and time	If the **Start type** is **Fixed**, specify the UTC start time for the return. The time should be at least 15 seconds in the future. Note that the time is the wall clock time, not the timecode in the input.

Field	Description
Reference action name	If the **Start type** is **Follow**, choose the input to follow. The dropdown list shows all existing input switches that are file inputs. Remember that a SCTE 35 action can follow input A only if input A is a file input and the source end behavior for input A is *continue*. For information about these switching rules, see the section called "Fixed, immediate, and follow switches".
Follow point	If the **Start type** is **Follow**, complete this field. The follow point is always **End**, to indicate that the switch will occur when the input in **Reference action name** has finished.
Splice event id	The ID of the splice_insert that the return-to-network should end. You assigned this ID when you created the splice_insert.

Fields for ID3 metadata

This table shows the fields that apply for an action to insert one ID3 metadata.

Field	Description
Action type	HLS Timed Metadata.
Action name	A name for the metadata item. You might want to design a convention for naming ID3 metadata items, such as **`id3_metadata-<UTC time>`**.
Start type	**Fixed** or **Immediate**.

Field	Description
Date and time	If the **Start type** is **Fixed**, specify the UTC start time for the ID3 metadata item. The time should be at least 15 seconds in the future. Note that the time is the wall clock time, not the timecode in the input.
Id3	Enter the ID3 metadata encoded as base64. The metadata must be fully formed ID3 metadata (including both a header and a frame, as per the ID3 2.4.0 specification). The content of the `value` property can include MediaLive variable data. MediaLive will examine the contents of the base64 and perform substitutions. For example, MediaLive will change **`$dt$`** to the date and time.

Fields for ID3 segment tags

This table shows the fields that apply for an action to insert ID3 segment tags.

Field	Description
Action type	HLS ID3 Segment Tagging.
Action name	A name for the segment tag.
Start type	**Fixed** or **Immediate**.
Date and time	If the **Start type** is **Fixed**, specify the UTC start time for the ID3 segment tag. The time should be at least 15 seconds in the future. Note that the time is the wall clock time, not the timecode in the input.

Field	Description
Tag	Complete this field if the content of the tag is free text. Enter the `value` for a `TXXX` field inside the ID3 tag. The content can include MediaLive variable data. In the following example, the content consists of the date and time, and the current segment number. The tag contents will be different in each segment. **`$dt$-$sn$`**
ID3	Complete this field if the content of the tag is ID3 metadata. Enter the content encoded as base64. The metadata must be fully formed ID3 metadata (including both a header and a frame, as per the ID3 specification).

Fields for pause

In **Schedule action settings**, complete the following fields.

Field	Description
Action type	Pause.
Action name	A name for the action.
Start type	**Fixed** or **Immediate**.
Date and time	If the **Start type** is **Fixed**, specify the UTC start time for the action. The time should be at least 15 seconds in the future. Note that the time is the wall clock time, not the timecode in the input.

Field	Description
Actions	Choose **Add actions**, then for **Pipeline id**, choose the pipeline that you want to pause: **PIPELINE_0** or **PIPELINE_1**.

When you choose **Create**, MediaLive adds an action to the schedule to pause the specified pipeline and to unpause any pipeline that isn't specified. As a result, only the specified pipeline will be paused after the action is performed.

Fields for unpause

In **Schedule action settings**, complete the following fields.

Field	Description
Action type	Pause.
Action name	A name for the action.
Start type	**Fixed** or **Immediate**.
Date and time	If the **Start type** is **Fixed**, specify the UTC start time for the action. The time should be at least 15 seconds in the future. Note that the time is the wall clock time, not the timecode in the input.
Actions	Keep this section empty. Don't add any actions.

When you choose **Create**, the empty **Actions** section instructs MediaLive to add an action to the schedule to unpause all pipelines.

Deleting actions from the schedule (console)

These rule apply when you add delete actions to the schedule:

- In a single-input channel (a channel that doesn't involve input switching), you can delete actions as follows:
  - You can delete an action with a start time that is more than 15 seconds in the future. The channel won't perform the action.
  - You can delete an action that has already been performed. Deleting this action doesn't reverse the action, it only removes it from the schedule.
- In a multiple-input channel, you can delete actions as follows:
  - You can delete an action with a start time that is more than 15 seconds in the future. The channel won't perform the action.

    There are some constraints that apply to deleting inputs switches and input prepare, event when they are in the future. For more information, see the section called “Deleting actions from the schedule” and the section called “Deleting and stopping”.
  - You can delete an action that has already been performed. Deleting this action doesn't reverse the action, it only removes it from the schedule.

    There are some constraints that apply to deleting inputs switches and input prepare, event when they are in the future. For more information, see the section called “Deleting actions from the schedule” and the section called “Deleting and stopping”.

Deleting versus reversing

It is important to understand that deleting a stale action from the schedule doesn't reverse its effect in the channel. For example, if you have paused the channel, and the channel has performed the action, you unpause the channel by entering a new action. You don't unpause it by deleting the action.

> **Note**
>
> If the channel has already received the action, you might be able to modify it to effectively delete it. For more information, see the section called “Modifying actions”.

You can delete any number of actions in one request, or any combination of types of actions in one request. For example, you can mix the deletion of SCTE 35 message actions and image overlay actions.

The general procedure is the same to delete any type of action.

To delete actions in list view

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Channel**, and then choose the channel that you want to work with.
3. On the **Details** pane, choose the **Schedule** tab.
4. If necessary, choose the **Switch** button to display the **List** view. For information about the layout and color coding of the timeline view, see the section called "Viewing the schedule".
5. Choose one or more actions to delete.

   If you choose an input switch that is in an input follow chain, a prompt appears. This prompt notifies you that all the follow input switch actions and the follow SCTE 35 actions up to the next fixed input switch will also be deleted. You can cancel or continue.

   Choose **Actions**, and then choose **Delete**.

To delete actions in timeline view (console)

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Channel**, and then choose the channel that you want to work with.
3. On the **Details** pane, choose the **Schedule** tab.
4. If necessary, choose the **Switch** button to display the **Timeline** view. For information about the layout and color coding of the timeline view, see the section called "Viewing the schedule".
5. In each action section, choose the **X** to delete the action.

   If you choose an input switch that is in an input follow chain, a prompt appears to notify you that the follow actions below this action (up to the next fixed input switch) will also be deleted. You can cancel or continue.

Modifying actions in the schedule (console)

You can't modify an action in the schedule, even if it hasn't been received by the channel. However, you can sometimes achieve a modify effect using a create action, a delete action, or both.

Topics

General rule about modifying actions

You can't modify an action in the schedule, even if it hasn't been received by the channel.

To change an action that hasn't yet started, delete the action and create it again. See the following sections for important tips on deleting and recreating.

Modifying an input switch action

You can't modify input switches in the schedule. But you can achieve the same result by deleting the action and creating it again.

Keep in mind that you can't delete or create an action that has a start time less than 15 seconds in the future. Give yourself enough time to delete and recreate the action before this deadline.

Modifying an input switch action in a follow chain

When you delete an action in an *input follow chain* (in order to delete and recreate it), you must also delete and recreate the input switch follow actions and SCTE 35 follow actions below this one. You must do this because each action refers to the previous action. If you delete the previous action, the next action becomes an orphan. Orphan actions aren't permitted.

Example 1: Modify an action

For example, assume this is the input follow chain:

```
Input A              Fixed    File
```

```
           Input B   Follow   File
           Input C   Follow   File
           Input D   Follow   File or Live
Input E              Fixed    File or Live
```

To delete and modify input A, you must also delete inputs B, C, and D. You must delete input B to prevent it becoming an orphan. The same rule applies until the next fixed input (input E), which isn't chained to another input. Therefore, you aren't required to delete input E.

When you delete input A using the console, a prompt appears to notify you that the follow actions below this action (up to input E, which is the next fixed input switch) will also be deleted. You can cancel or continue. You must then recreate inputs A to D. Recreate them in order going down the chain: input A, input B, input C, input D.

Example 2: Delete an action

This example shows how to delete input B:

```
Input A              Fixed    File
           Input B   Follow   File
           Input C   Follow   File
           Input D   Follow   File or Live
Input E              Fixed    File or Live
```

When you delete input B using the console, a prompt appears to notify you that the follow actions below this action (up to input E, which is the next fixed input switch) will also be deleted. You can cancel or continue. You must then recreate inputs C and D. Recreate them in order going down the chain: input C, input D. Remember to set up input C to follow input A instead of input B.

Inserting an input switch action into a follow chain

To insert a follow switch action into an *input follow chain* (between two existing follow switch actions), you must delete and recreate the follow actions below the insertion. You do that to prevent two actions from following one previous action. Branching is not allowed in the chain.

For example, suppose that you want to insert input X between input B and input C. Input C already refers to input B. You also want input X to refer to input B, but that is not allowed. Therefore, you

must delete input C and the inputs that follow. You then recreate the chain in the following order: input X (refers to input B), input C (refers to input X), input D (refers to input C).

Modifying an input prepare action

You can't modify input switches in the schedule. But you can achieve the same result by deleting the action and creating it again.

Keep in mind that you can't delete or create an action that has a start time less than 15 seconds in the future. Give yourself enough time to delete and recreate the action before this deadline.

Modifying a SCTE 35 message that is in progress

You can't modify a SCTE 35 message that is active in the channel. Specifically, you can't shorten the duration of a splice_insert. But you can achieve the same result by creating a return-to-network action.

Modifying or deleting an ID3 segment tag that is in progress

You can't modify the contents of an ID3 segment tag that is active in the channel. Instead, create a new action to override the active tag. The new action can contain a tag with new content, or it can contain an empty tag:

- If the new action contains a tag with content, the channel starts inserting the contents of the new action into every segment.
- If the new action contains an empty tag, the channel stops inserting segment tags, which means you have effectively deleted the tag.

Modifying an image overlay that is in progress

You can't directly modify an image overlay (either global or per-outputs) that is running in the channel. But you can achieve the same result by creating a new action with the same layer specified. You can do the following:

- Shorten or extend the duration of an image overlay.
- Change one or more attributes.
- Specify an attribute that isn't currently specified in an image overlay. For example, you might want to specify a fadeout where there is no fadeout in the current overlay.

- Create a new action (with a new action name) that inserts an image overlay 15 seconds in the future. Make sure to specify the following:
  - The same layer as the image that you want to modify.
  - A duration that is appropriate (the image doesn't inherit the duration of the current image).
  - All the attributes that you want.

  The new action replaces the current action because you specified the same layer.

Modifying a motion graphics overlay that is in progress

You can't modify a motion graphic overlay that is running in the channel. But you can achieve the same result in other ways. For example, the authoring system (that produces the motion graphic asset) can modify the content that is being published to the URL associated with the action. For information about preparing and publishing the motion graphics asset, see the section called “Step 1: Prepare the motion graphic asset”.

Viewing the schedule (console)

You can display the list of actions currently in the schedule and view them in list or timeline view.

To view actions (console)

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Channel**, and then choose the channel that you want to work with.
3. On the **Details** pane, choose the **Schedule** tab.

   The **Schedule actions** pane shows the actions in the schedule's current window of time.

   You can switch between the two views of the schedule by choosing the **Switch** button.

List view

The actions are mostly listed in chronological order.

Input switches in an input follow chain are grouped together starting from the top of the chain (which is the action above the first follow action) to the last follow action. Other actions, such as actions for SCTE 35 and image overlay, might occur between two follow actions. MediaLive can't

predict whether a SCTE 35 or image overlay action will occur between two follow actions and doesn't attempt to show it in the list view.

Timeline view

The actions are arranged in cards along a vertical axis. The card titles are color-coded by the action type. For example, SCTE 35 time_signal messages are yellow.

One card might contain several input switches. The first input switch is always a fixed-type input switch. The remaining input switches in that card are always follow-type input switches.

Working with the schedule (AWS CLI)

You can use the AWS CLI to work with the schedule programmatically. The sections later in this chapter describe how to enter the appropriate commands. These sections assume that you are familiar with the basics of using the AWS CLI. For information about the basics, see the AWS CLI Command Reference.

The following sections describe each command and provide this additional information:

- A description of the AWS CLI command syntax.
- A description of the schema for the request or response JSON payload. This payload is shown using the syntax for the AWS CLI.
- An example of the request or response JSON payload. This payload is also shown using the syntax for the AWS CLI.

For details on the JSON contents, we recommend that you read the AWS Elemental MediaLive API Reference. This guide is easy to use because it includes links from elements in the JSON payload to tables that describe the element. But you must adjust the syntax of the elements in the JSON code because the AWS CLI uses one form of capitalization for elements (for example, `SubSegmentNum`) and the API uses another form (for example, `subSegmentNum`).

Topics

- JSON payload for combining create and delete
- Viewing the schedule (AWS CLI)

Creating and deleting using a batch command

To create and delete actions in the schedule for a channel, you use the batch update schedule command. This command lets you perform multiple actions in one request. There isn't one command for creating actions and another for deleting actions.

> **Important**
>
> When working with a started and running channel, use the **batch-update-schedule** command to add or remove actions. Use the **delete-schedule** command only on idle channels. The **delete-schedule** command will delete all scheduled actions and could cause service interruptions if used on a live channel.

You can use the command as follows:

- Submit a *single* request such as a request to do the following:
  - Create one action.
  - Delete one action.
- Submit a *batch* request such as one request to do the following:
  - Create several actions.
  - Delete several actions.
  - Create one or more actions and delete one or more actions.

> **Important**
>
> In a command that combines create actions and delete actions, the delete actions are *always* performed before the create actions. This means that MediaLive removes the delete actions from the schedule before it adds the create actions to the schedule.

Topics

- How a batch request works
- Batch command in different interfaces
- JSON payload in different interfaces

How a batch request works

The intention of batching is to pass or fail all the actions together. Therefore, AWS Elemental MediaLive validates batch actions together. MediaLive performs the following validation:

- It ensures that each action that is created or deleted has an explicit or implied start time that is at least 15 seconds in the future.
- If an action refers to an existing action in the schedule, it ensures that the reference to the existing action is correct. For example, a follow input switch includes a reference to the action that it follows. That action must exist.

If the validation fails for any one action, it fails for all the actions in the batch.

If you don't want the actions to pass or fail together, don't submit a batch. Instead, create each action in its own batch update schedule command.

If the validation succeeds, MediaLive processes all the delete requests before the create requests, regardless of the start times of the actions.

Example 1

An important use of batching is to perform several actions that must pass or fail together. For example, suppose that you want to remove the corporate logo and immediately insert a splice_insert (in order to go to an ad avail). To do that, you must create an action to remove the logo and another action to insert the splice_insert. However, you don't want MediaLive to insert the remove action if the splice_insert action fails, or vice versa. It's better if both actions fail because that allows you to fix the badly formed action, and then submit both actions again.

You therefore submit the two actions together, in one batch update schedule command.

Example 2

Another important use of batching is to fix an error in an action in the schedule. For example, you might want to fix an image overlay that hasn't started yet and that was created with the wrong

start time. To do that, you submit one batch update schedule command with JSON that contains the following:

- A payload to remove the original action to activate the image overlay. This action has the incorrect start time.
- A payload to add a new action to activate the same image overlay. This action has the correct start time.

Batch command in different interfaces

The batch update schedule command is represented differently in different interfaces:

- In the AWS CLI, the command is `batch-update-schedule`.
- In the API, the command is represented by an `HTTP PUT` on `channels/channelId/schedule`.
- In the AWS SDKs, the command is represented by constructs that are suitable to that SDK language.

JSON payload in different interfaces

The JSON payload for the command is different for the different interfaces:

- In the AWS CLI, the contents of the payload depend on how you use the command:
  - You can enter a command with two parameters: `channel-id` and `--cli-input-json`. In this case, you create a file that repeats the channel ID and includes the JSON payload.
  - You can enter a command with three parameters: one for the channel ID, one for the JSON payload for the create actions (if applicable), and one for the JSON payload for the delete actions (if applicable). You pass the payloads in the command. If both parameters are present, each parameter takes a separate payload. But the two payloads are validated and performed as a batch.

  The payload for the AWS CLI is always pascal case (upper camel case).

- In the API, there is one payload with two sections, a `CREATES` section and a `DELETES` section. A request can contain one or both sections.

  The payload for the API is always camel case for variable names and pascal case for classes.

- In the AWS SDKs, the JSON payload is represented by constructs that are suitable to that SDK language.

To get more familiar with individual actions, we recommend that you use the MediaLive console to create an action. After you create the action, use the DescribeSchedule command in the appropriate interface (for example, the AWS CLI or an SDK) to obtain the raw JSON payload for the entire schedule. You can then copy individual actions and save them as models to use when working programmatically.

Submitting a batch update schedule command

The command for a batch update schedule command is identical for creating actions, deleting actions, or submitting a combination of create and delete actions. The command is identical. Only the contents of the JSON payload differ.

There are different ways to enter the command to create an action. We recommend that you follow this usage:

- Enter the command with two parameters: `channel-id` and `--cli-input-json`. In this case, you create a file that repeats the channel ID and includes the JSON payload.

The instructions and examples in this section illustrate this usage.

The following general rules apply to batch update commands:

- You can create actions when the channel is running or when it is idle.
- You can create any number of actions in one request, or any combination of types of actions in one request. For example, you can mix the creation of SCTE 35 message actions and image overlay actions.
- If you create several actions in one request and one of the create requests fails (usually because the start time isn't sufficiently in the future), then they all fail.

The following rules apply to delete actions:

- You can delete an action when the channel is running or when it is idle.

- You can delete any number of actions in one request, or any combination of types of actions in one request. For example, you can mix the deletion of SCTE 35 message actions and image overlay actions.
- If you delete several actions in one request and one of the delete requests fails (usually because the start time isn't sufficiently in the future), then they all fail.

To submit a batch command

1. Before you add or delete actions, read the section called “Creating actions” and the section called “Deleting actions”.
2. Prepare a file that contains the channel ID and the appropriate JSON payload for the actions. For the structure and examples of the JSON payload for different actions, see the sections that follow.
3. Give the file a suitable name with a `.txt` extension. For example, the file name for a payload that creates only actions might be `schedule-create-actions.txt`.
4. Save the file to the folder where you are running the AWS CLI.
5. On the command line, enter this command:

   ```
   AWS medialive batch-update-schedule --channel-id value --cli-input-json value
   ```

   - In the value for `--channel-id`, enter the channel ID as a number.
   - In the value for `--cli-input-json`, enter the file name in this format:

     ```
     file://filename.txt
     ```

   For example:

   ```
   aws medialive batch-update-schedule --channel-id 999999 --cli-input-json schedule-create-actions.txt
   ```
6. To submit the command, press **Enter**. The response appears on the screen. The response repeats the data from the request.

JSON payload for create actions

The following sections show the structure of the payload and an example of the payload for every type of create action for a MediaLive schedule.

Topics

Input switch action – payload

The following sections show the payload for input switch actions.

In this payload, the `ScheduleActionStartSettings` contains only one of `FixedModeScheduleActionStartSettings`, `ImmediateModeScheduleActionStartSettings`, or `FollowModeScheduleActionStartSettings`.

See the examples that follow for samples of each of these tags.

For information about the meaning and values for the fields in the following JSON, see the section called “Input switch”.

```
{
"ChannelId": "string",
"Creates": {
 "ScheduleActions": [
  {
   "ScheduleActionStartSettings": {
    "FixedModeScheduleActionStartSettings": {
     "Time": "string"
    },
    "FollowModeScheduleActionStartSettings": {
     "FollowPoint": "enum",
     "ReferenceActionName": "string"
    },
    "ImmediateModeScheduleActionStartSettings": {
    }
   },
   "ActionName": "string",
   "ScheduleActionSettings": {
    "InputSwitchSettings": {
     "InputAttachmentNameReference": "string",
     "InputClippingSettings": {
      "InputTimecodeSource": "enum",
      "StartTimecode": {
       "Timecode": "string"
      },
      "StopTimecode": {
       "LastFrameClippingBehavior": "enum",
       "Timecode": "string"
      }
     },
     "UrlPath": ["string", ...]
    }
   }
  ]
 }
}
```

Example of a switch to a live input with fixed start time

This example of a request is to switch to a live input at a fixed start time. The switch action is called `studio-feed` and it switches to the input that is connected to the input attachment called `live-studio-feed`. It switches to this input at the specified UTC time.

```
{
  "ChannelId": "999999",
  "Creates": {
    "ScheduleActions": [
      {
       "ScheduleActionStartSettings": {
          "FixedModeScheduleActionStartSettings": {
            "Time": "2018-05-21T20:42:19.000Z"
           }
          },
      "ActionName": "studio-feed",
      "ScheduleActionSettings": {
        "InputSwitchSettings": {
          "InputAttachmentNameReference": "live-studio-feed"
     }
    }
   }
  ]
 }
}
```

Example of a static file switch as a follow

This example of a request is to switch to a static file input to follow the end of the previous input. The switch action is called `action-ad-003` and it switches to the input that is connected to the input attachment called `zel-cafe`. It switches to this input when the action called `action-ad-002` ends. The file for this action is clipped so that it ends after 30 seconds and 11 frames:

```
{
  "ChannelId": "999999",
  "Creates": {
      "ScheduleActions": [
          {
            "ScheduleActionStartSettings": {
              "FollowModeScheduleActionStartSettings": {
                "FollowPoint": "END",
                "ReferenceActionName": "action-ad-002"
              }
            },
            "ActionName": "action-ad-003",
              "ScheduleActionSettings": {
                 "InputSwitchSettings": {
```

```
                    "InputAttachmentNameReference": "zel-cafe",
                      "InputClippingSettings": {
                          "InputTimecodeSource": "ZEROBASED",
                "StopTimecode":{
                  "Timecode": "00:00:30:11",
                  "LastFrameClippingBehavior": "INCLUDE_LAST_FRAME"
       }
      }
     }
    }
   }
  ]
 }
}
```

Example of a switch to a dynamic input with immediate start time

This example of a request is to switch to a dynamic file input immediately. The switch action is called `action-unscheduled-standby` and it switches to the input that is connected to the input attachment called `dynamic-unscheduled-standby`. For this usage of the dynamic input, the files to use are `oceanwaves.mp4`.

```
{
  "ChannelId": "999999",
  "Creates": {
    "ScheduleActions": [
      {
        "ScheduleActionStartSettings":
          {
          "ImmediateModeScheduleActionStartSettings": {
          }
         },
       "ActionName": "action-unscheduled-slate",
        "ScheduleActionSettings": {
          "InputSwitchSettings": {
            "InputAttachmentNameReference": "slate",
            "UrlPath":[
              "main/oceanwaves.mp4",
              "redundant/oceanwaves.mp4"]
     }
    }
   }
  ]
```

```
 }
}
```

Input prepare action – payload

The following sections show the payload for input switch actions.

In this payload, the `ScheduleActionStartSettings` contains only one of `FixedModeScheduleActionStartSettings`, `ImmediateModeScheduleActionStartSettings`, or `FollowModeScheduleActionStartSettings`.

See the examples that follow for samples of each of these tags.

For information about the meaning and values for the fields in the following JSON, see the section called “Input switch”.

```
{
"ChannelId": "string",
"Creates": {
 "ScheduleActions": [
  {
   "ScheduleActionStartSettings": {
    "FixedModeScheduleActionStartSettings": {
     "Time": "string"
    },
    "FollowModeScheduleActionStartSettings": {
     "FollowPoint": "enum",
     "ReferenceActionName": "string"
    },
    "ImmediateModeScheduleActionStartSettings": {
    }
   },
   "ActionName": "string",
   "ScheduleActionSettings": {
    "InputPrepareSettings": {
     "InputAttachmentNameReference": "string",
     "InputClippingSettings": {
      "InputTimecodeSource": "enum",
      "StartTimecode": {
       "Timecode": "string"
      },
      "StopTimecode": {
```

```
        "LastFrameClippingBehavior": "enum",
        "Timecode": "string"
       }
      },
      "UrlPath": ["string", ...]
      }
     }
    }
   ]
  }
}
```

Example of an input prepare with a fixed start time

This example of a request is to switch to a live input at a fixed start time. The switch action is called `studio-feed` and it switches to the input that is connected to the input attachment called `live-studio-feed`. It switches to this input at the specified UTC time.

```
{
  "ChannelId": "999999",
  "Creates": {
    "ScheduleActions": [
      {
       "ScheduleActionStartSettings": {
          "FixedModeScheduleActionStartSettings": {
            "Time": "2018-05-21T20:42:19.000Z"
           }
          },
      "ActionName": "studio-feed",
      "ScheduleActionSettings": {
        "InputSwitchSettings": {
          "InputAttachmentNameReference": "live-studio-feed"
     }
    }
   }
  ]
 }
}
```

Example of an input prepare as a follow

This example of a request is to switch to a static file input to follow the end of the previous input. The switch action is called `action-ad-003` and it switches to the input that is connected to the

input attachment called `zel-cafe`. It switches to this input when the action called `action-ad-002` ends. The file for this action is clipped so that it ends after 30 seconds and 11 frames:

```
{
  "ChannelId": "999999",
  "Creates": {
      "ScheduleActions": [
          {
            "ScheduleActionStartSettings": {
              "FollowModeScheduleActionStartSettings": {
                "FollowPoint": "END",
                "ReferenceActionName": "action-ad-002"
              }
            },
            "ActionName": "action-ad-003",
              "ScheduleActionSettings": {
                 "InputSwitchSettings": {
                    "InputAttachmentNameReference": "zel-cafe",
                      "InputClippingSettings": {
                          "InputTimecodeSource": "ZEROBASED",
               "StopTimecode":{
                 "Timecode": "00:00:30:11",
                 "LastFrameClippingBehavior": "INCLUDE_LAST_FRAME"
              }
            }
          }
        }
      }
    ]
  }
}
```

Activate global image action – payload

For information about the meaning and values for the fields in the following JSON, see the section called “Activate global image overlay”.

```
{
"ChannelId": "string",
"Creates": {
 "ScheduleActions": [
  {
```

```
    "ScheduleActionStartSettings": {
     "FixedModeScheduleActionStartSettings": {
      "Time": "string"
     },
     "ImmediateModeScheduleActionStartSettings": {
     }
    },
    "ActionName": "string",
    "ScheduleActionSettings": {
     "StaticImageActivateSettings": {
      "Duration": integer,
      "FadeIn": integer,
      "FadeOut": integer,
      "Height": integer,
      "Image": {
       "PasswordParam": "string",
       "Uri": "string",
       "Username": "string"
      },
      "ImageX": integer,
      "ImageY": integer,
      "Layer": integer,
      "Opacity": integer,
      "Width": integer
     }
    }
   }
  ]
 }
}
```

Example

This example of a request creates an image overlay to be inserted in every video output in every output group in the channel. The overlay uses a file that is stored in an Amazon S3 bucket. The request doesn't include a duration and therefore doesn't include a fadeout. Instead, the intention is to send a separate deactivate request at the appropriate time. All the times are in milliseconds, and all the positioning values are in pixels:

```
{
  "ChannelId": "999999",
  "Creates": {
  "ScheduleActions": [
```

```
    {
      "ScheduleActionStartSettings": {
        "FixedModeScheduleActionStartSettings": {
          "Time": "2018-05-21T20:42:19.000Z"
        }
      },
      "ActionName": "corporate-logo-030",
      "ScheduleActionSettings": {
        "StaticImageActivateSettings": {
          "Image": {
          "PasswordParam": "corplogo!2312",
          "Uri": "s3ssl://DOC-EXAMPLE-BUCKET/logos/corporate/high-res.bmp",
          "Username": "medialiveoperator"
          },
          "Layer": 1,
          "FadeIn": 1500,
          "Height": 900
          "Width": 800,
          "ImageX": 200,
          "ImageY": 300,
          "Opacity": 60,
          }
        }
      }
    ]
  }
}
```

Deactivate global overlay action – payload

For information about the meaning and values for the fields in the following JSON, see the section called “Deactivate global image overlay”.

```
{
 "ChannelId": "string",
 "Creates": {
 "ScheduleActions": [
  {
   "ScheduleActionStartSettings": {
    "FixedModeScheduleActionStartSettings": {
     "Time": "string"
    },
    "ImmediateModeScheduleActionStartSettings": {
```

```
    }
   },
   "ActionName": "string",
   "ScheduleActionSettings": {
    "StaticImageDeactivateSettings": {
     "FadeOut": integer,
     "Layer": integer
    }
   }
  }
 ]
}
```

Example

This example of a request creates an action to remove an image overlay at 20:42:04.000 (UTC) with a 500-millisecond fadeout that is added onto the end time, which means that the overlay will be invisible at 20:42:04.500.

The action removes the images overlay that is in the global layer 4. This means it removes the image *only if* it was inserted using the global action (StaticImageActivateSettings). It doesn't remove the overlay from the per-outputs layer 4.

```
{
  "ChannelId": "999999",
  "Creates": {
  "ScheduleActions": [
    {
      "ScheduleActionStartSettings": {
        "FixedModeScheduleActionStartSettings": {
          "Time": "2018-05-21T20:42:04.000Z"
        }
      },
      "ActionName": "stop-overlay-029",
      "ScheduleActionSettings": {
        "StaticImageDeactivateSettings": {
          "FadeOut": 500,
          "Layer": 4
          }
        }
      }
    ]
  }
```

```
}
```

Activate per-outputs image action – payload

For information about the meaning and values for the fields in the following JSON, see the section called "Activate per-outputs image overlay".

```
{
"ChannelId": "string",
"Creates": {
 "ScheduleActions": [
  {
   "ScheduleActionStartSettings": {
    "FixedModeScheduleActionStartSettings": {
     "Time": "string"
    },
    "ImmediateModeScheduleActionStartSettings": {
    }
   },
   "ActionName": "string",
   "ScheduleActionSettings": {
    "StaticImageOutputActivateSettings": {
     "Duration": integer,
     "FadeIn": integer,
     "FadeOut": integer,
     "Height": integer,
     "Image": {
      "PasswordParam": "string",
      "Uri": "string",
      "Username": "string"
     },
     "ImageX": integer,
     "ImageY": integer,
     "Layer": integer,
     "Opacity": integer,
     "OutputNames": [
       {}
      ],
     "Width": integer
     }
    }
   }
  ]
```

```
 }
}
```

Example

This example of a request creates an image overlay to be inserted in specific outputs in the channel. The overlay uses a file that is stored in an Amazon S3 bucket. The request inserts the image in the outputs `hls-high-res` and `mss-high-res`. The image is sized appropriately for the video resolution in these outputs.

The request is intended to be always present. Therefore, it is set up to start immediately (as soon as the channel starts)All the times are in milliseconds, and all the positioning values are in pixels:

```
{
  "ChannelId": "999999",
  "Creates": {
  "ScheduleActions": [
    {
        "ScheduleActionStartSettings":
          {
          "ImmediateModeScheduleActionStartSettings": {
          }
         },
      },
      "ActionName": "logo-1280",
      "ScheduleActionSettings": {
        "StaticImageOutputActivateSettings": {
          "Image": {
          "PasswordParam": "corplogo!2312",
          "Uri": "s3ssl://DOC-EXAMPLE-BUCKET/logos/corporate/10percent-1280.bmp",
          "Username": "medialiveoperator"
          },
          "Layer": 0,
          "outputNames": [
            hls-high-res,mss-high-res
           ],
          "ImageX": 200,
          "ImageY": 300,
          "FadeIn": 1500,
          "Opacity": 60
          }
        }
      }
```

```
    ]
  }
}
```

Deactivate per-outputs overlay action – payload

For information about the meaning and values for the fields in the following JSON, see the section called “Deactivate per-outputs image overlay”.

```
{
 "ChannelId": "string",
 "Creates": {
 "ScheduleActions": [
  {
   "ScheduleActionStartSettings": {
    "FixedModeScheduleActionStartSettings": {
     "Time": "string"
    },
    "ImmediateModeScheduleActionStartSettings": {
    }
   },
   "ActionName": "string",
   "ScheduleActionSettings": {
    "StaticImageOutputDeactivateSettings": {
     "FadeOut": integer,
     "Layer": integer,
     "OutputNames"
    }
   }
  }
 ]
}
```

Example

The deactivate action deactivates the image in the specified per-output layer, in the specified output or outputs.

In this example, which follows on from the activate example, the action removes all image overlays that are in per-outputs layer 4 in the output hls-high-res.

```
{
  "ChannelId": "999999",
```

```
  "Creates": {
  "ScheduleActions": [
    {
      "ScheduleActionStartSettings": {
        "FixedModeScheduleActionStartSettings": {
          "Time": "2018-05-21T20:42:04.000Z"
        }
      },
      "ActionName": "stop-layer4-all-outputs",
      "ScheduleActionSettings": {
        "StaticImageOutputDeactivateSettings": {
          "outputNames": [
            hls-high-res
           ],
          "FadeOut": 500,
          "Layer": 4
          }
        }
      }
    ]
  }
}
```

Activate motion graphic overlay – payload

For information about the meaning and values for the fields in the following JSON, see the section called “Activate motion graphics overlay”.

```
{
 "ChannelId": "string",
 "Creates": {
 "ScheduleActions": [
  {
   "ScheduleActionStartSettings": {
    "FixedModeScheduleActionStartSettings": {
     "Time": "string"
    },
    "ImmediateModeScheduleActionStartSettings": {
    }
   },
   "ActionName": "string",
   "ScheduleActionSettings": {
    "MotionGraphicsImageActivateSettings": {
```

```
      "Duration": integer
      "Url": "string"
      "Username": "string",
      "PasswordParam": "string"
     }
    }
   ]
  }
}
```

Example

This example of a request creates a motion graphics overlay action called mg_ticker_tape. The motion graphic asset is stored at http://example.com/ticker_tape.html. This server requires user credentials. The request doesn't include a duration. Instead, the intention is to send a separate deactivate request at the appropriate time.

```
{
"ChannelId": "999999",
"Creates": {
 "ScheduleActions": [
  {
   "ScheduleActionStartSettings": {
    "FixedModeScheduleActionStartSettings": {
     "Time": "2018-05-21T20:42:04.000Z"
    }
   },
   "ActionName": "mg_ticker_tape",
   "ScheduleActionSettings": {
    "MotionGraphicsImageActivateSettings": {
     "Url": "https://example.com/ticker_tape.html"
     "Username": "medialiveoperator",
     "PasswordParam": "/medialive/12345"
    }
   }
  ]
 }
}
```

Deactivate motion graphic overlay – payload

For information about the meaning and values for the fields in the following JSON, see the section called “Deactivate motion graphics overlay”.

```
{
"ChannelId": "string",
"Creates": {
 "ScheduleActions": [
  {
   "ScheduleActionStartSettings": {
    "FixedModeScheduleActionStartSettings": {
     "Time": "string"
    },
    "ImmediateModeScheduleActionStartSettings": {
    }
   },
   "ActionName": "string",
   "ScheduleActionSettings": {
    "MotionGraphicsImageDeactivateSettings": {
     }
    }
   }
  ]
 }
}
```

Example

This example of a request creates an action to end a motion graphic overlay at 23:59:00.000 (UTC). :

```
{
"ChannelId": "999999",
"Creates": {
 "ScheduleActions": [
  {
  "ScheduleActionStartSettings": {
   "FixedModeScheduleActionStartSettings": {
   "Time": "2018-05-21T23:59:00.000Z"
   },
   "ActionName": "deactivate-ticker-tape",
   "ScheduleActionSettings": {
```

```
    "MotionGraphicsImageDeactivateSettings": {
     }
    }
   }
  ]
 }
}
```

Splice_insert message – payload

For information about the meaning and values for the fields in the following JSON, see see the section called "Splice_insert".

```
{
 "ScheduleActions": [
  {
   "ScheduleActionStartSettings": {
    "FixedModeScheduleActionStartSettings": {
     "Time": "string"
    },
    "FollowModeScheduleActionStartSettings": {
     "FollowPoint": "enum",
     "ReferenceActionName": "string"
    },
    "ImmediateModeScheduleActionStartSettings": {
    }
   },
   "ActionName": "string",
   "ScheduleActionSettings": {
    "Scte35SpliceInsertSettings": {
     "Duration": integer,
     "SpliceEventId": integer
    }
   }
  }
 ]
}
```

Example of a splice insert with a fixed start time

This example of a request creates an action for a splice_insert with a UTC start time of 20:42:04.000. It also has an `ActionName` that perhaps references an ad avail from your database, a unique integer for the splice event ID, and a duration of 1,350,000 kHz ticks (15 seconds).

```
{
  "ChannelId": "999999",
  "Creates": {
    "ScheduleActions": [
      {
        "ScheduleActionStartSettings": {
          "FixedModeScheduleActionStartSettings": {
            "Time": "2018-05-21T20:42:04.000Z"
          }
        },
      "ActionName": "adavail-3708",
      "ScheduleActionSettings": {
        "Scte35SpliceInsertSettings": {
          "SpliceEventId": 3708,
          "Duration": 1350000
          }
        }
      }
    ]
  }
}
```

Example of a splice insert as a follow

This example of a request creates an action for a splice_insert to be inserted after the input switch called nature-doco-003 ends. The action has an `ActionName` that perhaps references an ad avail from your database, a unique integer for the splice event ID, and a duration of 1,350,000 kHz ticks (15 seconds).

Follow mode for a SCTE 35 message is useful when you want an ad avail to occur as soon as an input finishes, but you don't know when that will happen.

```
{
  "ChannelId": "999999",
  "Creates": {
    "ScheduleActions": [
      {
        "ScheduleActionStartSettings": {
         "FollowModeScheduleActionStartSettings": {
                "FollowPoint": "END",
                "ReferenceActionName": "nature-doco-003"
        }
```

```
      },
      "ActionName": "adavail-3708",
      "ScheduleActionSettings": {
        "Scte35SpliceInsertSettings": {
          "SpliceEventId": 3708,
          "Duration": 1350000
          }
        }
      }
    ]
  }
}
```

Time_signal message – payload

For information about the meaning and values for the fields in the following JSON, see the section called “Time_signal”.

```
{
 "ScheduleActions": [
  {
   "ScheduleActionStartSettings": {
    "FixedModeScheduleActionStartSettings": {
     "Time": "string"
    },
    "FollowModeScheduleActionStartSettings": {
     "FollowPoint": "enum",
     "ReferenceActionName": "string"
    },
    "ImmediateModeScheduleActionStartSettings": {
    }
   },
   "ActionName": "string",
   "ScheduleActionSettings": {
    "Scte35TimeSignalSettings": {
     "Scte35Descriptors": [
      {
       "Scte35DescriptorSettings": {
        "SegmentationDescriptorScte35DescriptorSettings": {
         "DeliveryRestrictions": {
          "ArchiveAllowedFlag": "enum",
          "DeviceRestrictions": "enum",
          "NoRegionalBlackoutFlag": "enum",
```

```
          "WebDeliveryAllowedFlag": "enum"
         },
         "SegmentNum": integer,
         "SegmentationCancelIndicator": "enum",
         "SegmentationDuration": integer,
         "SegmentationEventId": integer,
         "SegmentationTypeId": integer,
         "SegmentationUpid": "string",
         "SegmentationUpidType": integer,
         "SegmentsExpected": integer,
         "SubSegmentNum": integer,
         "SubSegmentsExpected": integer
        }
       }
      }
     ]
    }
   }
  }
 ]
}
```

Example

This example of a request creates an action for a time_signal with a UTC start time of 20:42:04.000 and with a unique integer for `SegmentationEventId`. For the restrictions fields, `NoRegionalBlackoutFlag` has a restriction set (regional blackouts are in place).

```
{
  "ChannelId": "999999",
  "Creates": {
  "ScheduleActions": [
    {
      "ScheduleActionStartSettings": {
        "FixedModeScheduleActionStartSettings": {
          "Time": "2018-05-21T20:42:04.000Z"
        }
      },
      "ActionName": "adavail-3708",
      "ScheduleActionSettings": {
        "Scte35TimeSignalSettings": {
          "Scte35Descriptors": [
          {
```

```
            "Scte35DescriptorSettings": {
              "SegmentationDescriptorScte35DescriptorSettings": {
                "SubSegmentsExpected": 0,
                "SegmentationEventId": 7054,
                "SegmentationDuration": 1350000,
                "SegmentationCancelIndicator": 0,
                "SubSegmentNum": 0,
                "SegmentationUpidType": 12,
                "SegmentNum": 0,
                "SegmentationCancelIndicator": "SEGMENTATION_EVENT_NOT_CANCELED",
                "DeliveryRestrictions": {
                  "DeviceRestrictions": "NONE",
                  "WebDeliveryAllowedFlag": "WEB_DELIVERY_ALLOWED",
                  "NoRegionalBlackoutFlag": "REGIONAL_BLACKOUT",
                  "ArchiveAllowedFlag": "ARCHIVE_ALLOWED"
                },
                "SegmentationUpid": "4a414e3136494e4155303031",
                "SegmentationTypeId": 52,
                "SegmentsExpected": 0
                  }
                }
              }
            ]
          }
        }
      }
    ]
  }
}
```

Return-to-network message – payload

For information about the meaning and values for the fields in the following JSON, see the section called “Return-to-network”.

```
{
 "ScheduleActions": [
  {
   "ScheduleActionStartSettings": {
    "FixedModeScheduleActionStartSettings": {
     "Time": "string"
    },
    "FollowModeScheduleActionStartSettings": {
```

```
     "FollowPoint": "enum",
     "ReferenceActionName": "string"
    },
    "ImmediateModeScheduleActionStartSettings": {
    }
   },
   "ActionName": "string",
   "ScheduleActionSettings": {
    "Scte35ReturnToNetworkSettings": {
     "SpliceEventId": integer
    }
   }
  }
 ]
}
```

Example

This example of a request creates a return-to-network with a UTC start time of 20:42:19.

```
{
  "ChannelId": "999999",
  "Creates": {
    "ScheduleActions": [
      {
        "ScheduleActionStartSettings": {
          "FixedModeScheduleActionStartSettings": {
            "Time": "2018-05-21T20:42:19.000Z"
          }
      },
      "ActionName": "end-adavail-3708",
      "ScheduleActionSettings": {
        "Scte35ReturnToNetworkSettings": {
          }
        }
      }
    ]
  }
}
```

ID3 metadata item – payload

For information about the meaning and values for the fields in the following JSON, see the section called “ID3 metadata”.

```
{
 "ScheduleActions": [
  {
   "ScheduleActionStartSettings": {
    "FixedModeScheduleActionStartSettings": {
     "Time": "string"
    },
    "ImmediateModeScheduleActionStartSettings": {
    }
   },
   "ActionName": "string",
   "ScheduleActionSettings": {
 "HlsId3SegmentTaggingSettings": {
     "Tag": "string"
    },
    "HlsTimedMetadataSettings": {
     "Id3": "string"
    }
   }
  }
 ]
}
```

Example

This example of a request creates ID3 metadata to be inserted at 13:35:59 UTC.

```
{
  "ChannelId": "999999",
  "Creates": {
  "ScheduleActions": [
    {
      "ScheduleActionStartSettings": {
        "FixedModeScheduleActionStartSettings": {
          "Time": "2019-01-02T13:35:59Z"
        }
      },
      "ActionName": "id3-metadata.2019-01-02T13:35:59Z",
```

```
      "ScheduleActionSettings": {
        "HlsTimedMetadataSettings": {
          "Id3": "SUQzBAAAAAAAF1RJVDIAAAANAAADSGVsbG8gV29ybGQA"
          }
        }
      }
    ]
  }
}
```

ID3 segment tag item – payload

For information about the meaning and values for the fields in the following JSON, see the section called “ID3 segment tags”.

```
{
 "ScheduleActions": [
  {
   "ScheduleActionStartSettings": {
    "FixedModeScheduleActionStartSettings": {
     "Time": "string"
    },
    "ImmediateModeScheduleActionStartSettings": {
    }
   },
   "ActionName": "string",
   "ScheduleActionSettings": {
 "HlsId3SegmentTaggingSettings": {
     "Tag": "string"
     "Id3": "string"
    }
   }
  }
 ]
}
```

Example using the tag property

This example of a request creates an ID3 segment tag to be inserted starting at 13:35:59 UTC. This example uses the `Tag` property of `HlsId3SegmentTaggingSettings`, which means that you specify only the content of the `value` for the TXXX field. In this example, the content is the date, time, and number of the segment.

```
{
  "ChannelId": "999999",
  "Creates": {
  "ScheduleActions": [
    {
      "ScheduleActionStartSettings": {
        "FixedModeScheduleActionStartSettings": {
          "Time": "2020-01-02T13:35:59Z"
        }
      },
      "ActionName": "id3-datetime-and-segment",
      "ScheduleActionSettings": {
        "HlsId3SegmentTaggingSettings": {
          "Tag": "$dt$-$sn$"
          }
        }
      }
    ]
  }
}
```

Example using the Id3 property

This example of a request creates an ID3 segment tag to be inserted immediately. This example uses the `Id3` property of `HlsId3SegmentTaggingSettings`, which means that the content is encoded as base64.

```
{
  "ChannelId": "999999",
  "Creates": {
  "ScheduleActions": [
    {
      "ScheduleActionStartSettings": {
        ImmediateModeScheduleActionStartSettings
       }
      },
      "ActionName": "id3-song309",
      "ScheduleActionSettings": {
        "HlsId3SegmentTaggingSettings": {
          "Id3": "SUQzBAAAAAAAF1RJVDIAAAANAAADSGVsbG8gV29ybGQA"
          }
        }
      }
```

```
    ]
  }
}
```

Pause pipeline action – payload

For information about the meaning and values for the fields in the following JSON, see the section called “Pause”.

```
{
 "ScheduleActions": [
  {
   "ScheduleActionStartSettings": {
    "FixedModeScheduleActionStartSettings": {
     "Time": "string"
    },
    "ImmediateModeScheduleActionStartSettings": {
    }
   },
   "ActionName": "string",
   "ScheduleActionSettings": {
    "PauseStateSettings": {
     "Pipelines": [
      {
       "PipelineId": "enum"
      }
     ]
    }
   }
  }
 ]
}
```

Example: Pausing one pipeline

This example of a request pauses pipeline 0 at 20:42:19 UTC. MediaLive always reads the command as:*set the specified pipeline or pipelines to pause and set all other pipelines to unpaused.*

```
{
  "ChannelId": "999999",
  "Creates": {
    "ScheduleActions": [
      {
```

```
        "ScheduleActionStartSettings": {
          "FixedModeScheduleActionStartSettings": {
            "Time": "2018-05-21T20:42:19Z"
          }
        },
        "ActionName": "pause-pipeline-0-now",
        "ScheduleActionSettings": {
          "PauseStateSettings": {
            "Pipelines": [
              {
                "PipelineId": "PIPELINE_0"
              }
            ]
          }
        }
      }
    ]
  }
}
```

Example: Unpausing both pipelines

This example of a request unpauses all pipelines that are currently paused.

> **Note**
>
> MediaLive always reads the command as:*set the specified pipeline or pipelines to pause and set all other pipelines to unpaused.* In this example, the `Pipelines` array is empty. MediaLive interprets this empty array as: *set all pipelines to unpaused*.

```
{
 "ChannelId": "999999",
 "Creates": {
     "ScheduleActions": [
      {
       "ScheduleActionStartSettings": {
         "ImmediateModeScheduleActionStartSettings": {}
      },
     "ActionName": "unpause-pipeline-0",
     "ScheduleActionSettings": {
       "PauseStateSettings": {
```

```
          "Pipelines": [
        {}
       ]
      }
     }
    }
   ]
  }
}
```

Combination of create actions

Here is an example of a JSON body to pass into the `--creates` parameter of the `batch-update-schedule` AWS CLI command. It contains two actions to create. In this example, both actions are splice_inserts, but in fact you can combine any number and any type of create actions.

```
{
    "ScheduleActions": [
      {
        "ScheduleActionSettings": {
          "Scte35SpliceInsertSettings": {
            "Duration": 1350000,
            "SpliceEventId": 3
          }
        },
        "ActionName": "SpliceInsert-01",
        "ScheduleActionStartSettings": {
          "FixedModeScheduleActionStartSettings": {
            "Time": "2018-11-05T16:10:30.000Z"
          }
        }
      },
      {
        "ScheduleActionSettings": {
          "Scte35SpliceInsertSettings": {
            "Duration": 2700000,
            "SpliceEventId": 3
          }
        },
        "ActionName": "SpliceInsert-02",
        "ScheduleActionStartSettings": {
          "FixedModeScheduleActionStartSettings": {
            "Time": "2018-11-05T16:30:45.000Z"
```

```
          }
        }
      }
    ]
  }
```

JSON payload for delete actions

In the `Deletes` section, include the list of actions to delete by entering an array of `ActionNames`. The array contains one or more action names. You can obtain these action names using the `DescribeChannel` command (see the section called “Viewing the schedule”.

```
{
  "ChannelId": "string",
  "Deletes": {
      "ActionNames": [
          ""
    ]
  }
}
```

Example

This example of a request deletes the three actions identified by `ActionNames` that were assigned when you created the actions.

```
{
  "ChannelId": "999999",
  "Deletes": {
      "ActionNames": [
          "stop-overlay-33",
          "adavail-3711",
          "end-adavail-3711"
      ]
  }
}
```

JSON payload for combining create and delete

To combine a batch of creates and deletes, include both a `Creates` section and a `Deletes` section in the JSON payload.

In this example, the payload in the `Deletes` section removes an action to activate an image overlay because it has an incorrect start time. The action is named `overlay-21`. The payload in the `Creates` section inserts that action again, this time with the correct start time.

Even though the `Creates` section appears first in the JSON payload, MediaLive always performs the delete actions first.

In this action, the delete action and the create action have the same `ActionName`. The name is being reused because the batch is a "delete and replace." But you could assign a different name to the create action.

```
{
  "ChannelId": "999999",
  "Creates": {
  "ScheduleActions": [
    {
      "ScheduleActionStartSettings": {
        "FixedModeScheduleActionStartSettings": {
          "Time": "2018-05-21T20:42:19.000Z"
        }
      },
      "ActionName": "overlay-21",
      "ScheduleActionSettings": {
        "StaticImageActivateSettings": {
          "Image": {
          "PasswordParam": "imagespassword",
          "Uri": "s3ssl://DOC-EXAMPLE-BUCKET/banner-A/high-res.bmp",
          "Username": "medialiveoperator"
          },
          "ImageY": 300,
          "FadeIn": 1500,
          "ImageX": 200,
          "Width": 800,
          "Opacity": 60,
          "Layer": 1,
          "Height": 900
          }
        }
      }
    ]
  },
  "Deletes": {
       "ActionNames": [
```

```
            "overlay-21"
        ]
    }
}
```

Viewing the schedule (AWS CLI)

You can use the AWS CLI to view a list of the actions that are currently in the schedule for one channel:

- Actions that have not yet been performed in the channel
- Actions that have previously been performed

To view the schedule, use the `DescribeSchedule` command. This command is represented differently in different interfaces:

- In the AWS CLI, the command is `describe-schedule`.
- In the API, the command is represented by an `HTTP GET` on `channels/channelId/schedule`.
- In the AWS SDKs, the command is represented by constructs that are suitable to that SDK language.

To view actions (AWS CLI)

1. Enter this command:

   aws medialive describe-schedule --channel-id *value* --max-results value
2. To submit the command, press **Enter**. The response appears on the screen.
3. If you used the `-max-results` option and the response included `NextToken`, enter the **DescribeChannel** command and pass the value of `NextToken` in `--next-token`. For example:

   aws medialive describe-schedule --channel-id *value* --next-token 3jhrprd0
4. To submit the command, press **Enter**. The response appears on the screen.

Example

The JSON body of the command *response* is similar to that of the `BatchUpdateSchedule` command *request*.

This example of a response shows the following actions:

- An action with the `ActionName` **`corporate-logo-029`** to activate an image overlay in layer 1 at 20:30:00 UTC
- An action with the `ActionName` **`stop-overlay-029`** to deactivate the overlay in layer 1 at 20:42:04 UTC
- An action with the `ActionName` **`adavail-3708`** to insert a splice_insert at the same time as the deactivate action
- An action with the `ActionName` **`end-adavail-3708`** to return-to-network 15 seconds later, at 20:42:19 UTC
- An action with the `ActionName` **`corporate-logo-030`** to reactivate the same overlay in layer 1 at the same time as the return

This schedule describes a workflow where you generally show your corporate logo, but you remove it at the start of each ad avail and then display it again at the end of the ad avail.

```
{
 "NextToken": "3jhrprd0",
     "ScheduleActions": [
     {
       "ScheduleActionStartSettings": {
         "FixedModeScheduleActionStartSettings": {
           "Time": "2018-05-21T20:30:00.000Z"
         }
       },
       "ActionName": "corporate-logo-029",
       "ScheduleActionSettings": {
         "StaticImageActivateSettings": {
           "Image": {
           "PasswordParam": "corplogo!2312",
           "Uri": "s3ssl://DOC-EXAMPLE-BUCKET/logos/corporate/high-res.bmp",
           "Username": "medialiveoperator"
           },
           "ImageY": 300,
           "FadeIn": 1500,
```

```
            "ImageX": 200,
            "Width": 800,
            "Opacity": 60,
            "Layer": 1,
            "Height": 900
          }
        }
      },
      {
        "ScheduleActionStartSettings": {
          "FixedModeScheduleActionStartSettings": {
            "Time": "2018-05-21T20:42:04.000Z"
          }
        },
        "ActionName": " stop-overlay-029",
        "ScheduleActionSettings": {
          "StaticImageDeactivateSettings": {
            "FadeOut": 1500,
            "Layer": 1
          }
        }
      },
      {
        "ScheduleActionStartSettings": {
          "FixedModeScheduleActionStartSettings": {
            "Time": "2018-05-21T20:42:04.000Z"
          }
        },
        "ActionName": "adavail-3708",
        "ScheduleActionSettings": {
          "Scte35SpliceInsertSettings": {
            "SpliceEventId": 3708,
            "Duration": 1350000
          }
        }
      },
      {
        "ScheduleActionStartSettings": {
          "FixedModeScheduleActionStartSettings": {
            "Time": "2018-05-21T20:42:19.000Z"
          }
        },
        "ActionName": "end-adavail-3708",
        "ScheduleActionSettings": {
```

```
          "Scte35ReturnToNetworkSettings": {
            "SpliceEventId": 3708
          }
        }
      },
      {
        "ScheduleActionStartSettings": {
          "FixedModeScheduleActionStartSettings": {
            "Time": "2018-05-21T20:42:19.000Z"
          }
        },
        "ActionName": "corporate-logo-030",
        "ScheduleActionSettings": {
          "StaticImageActivateSettings": {
            "Image": {
            "PasswordParam": "corplogo!2312",
            "Uri": "s3ssl://DOC-EXAMPLE-BUCKET/logos/corporate/high-res.bmp",
            "Username": "medialiveoperator"
            },
            "ImageY": 300,
            "FadeIn": 1500,
            "ImageX": 200,
            "Width": 800,
            "Opacity": 60,
            "Layer": 1,
            "Height": 900
          }
        }
      }
    ]
  }
```

Starting, stopping, and pausing an AWS Elemental MediaLive channel

After you create a channel, you can start it. The channel never starts automatically except when it is already running and attempts to recover from a failure.

You can stop a running channel at any time.

You can also pause one or both the pipelines in a channel by adding a Pause action to the schedule for the channel. For more information, see the section called “Pause and unpause”.

For information about charges for a channel, see the section called “Pricing”. There are different charges depending on the state of the channel:

- Charges when the channel is running
- Charges when the channel is idle

To start a channel

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Channels**, and then on the **Channels** page, choose the channel that you want to start.
3. Before you start the channel, decide if you want to enable thumbnails for the channel. You won't be able to enable thumbnails after the channel starts.
4. Choose **Start**. The channel state changes to one of the following:
   - **Starting**
   - **Running** (encoding on the pipeline or pipelines)
5. Choose the channel name. The details for the channel appear. After a few seconds, the thumbnail preview of the current input appears (if thumbnail preview is enabled).

To stop a channel

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Channels**, and then on the **Channels** page, choose the channel that you want to stop.

3. Choose **Stop**.

   The thumbnail preview (if thumbnails preview is enabled in the channel) stops updating. After a few seconds, the current thumbnail is replaced by a message.

Monitoring channel activity in AWS Elemental MediaLive

You can monitor activity on channels and multiplexes on the AWS Elemental MediaLive console, or using Amazon CloudWatch Events, Amazon CloudWatch Logs, or AWS CloudTrail.

Topics

Types of activity

The following table summarizes the type of activity you can monitor, and the service you can use.

Activity		MediaLive console	CloudWatch events	CloudWatch metrics	CloudWatch logs	CloudTrail events
State of channel	Report state of a channel	Yes	Yes		Yes	
State of multiplex	Report state of a multiplex	Yes	Yes		Yes	
Alerts	Generate alerts when a channel is running	Yes	Yes	Yes (count of active alerts)	Yes	

Activity		MediaLive console	CloudWatch events	CloudWatch metrics	CloudWatch logs	CloudTrail events
Metrics	Generate metrics			Yes		
Logs for channel	Log activity when a channel is running				Yes	
Logs for schedule	Log active schedule actions				Yes	
Logs for API calls	Log API calls, including those performed from the console				Yes	Yes

The following sections provide details about some of these types of activities.

Topics

States for channels

MediaLive reports the state of all channels.

MediaLive turns these states into CloudWatch events with the detailType set to `MediaLive Channel State Change`. For an example of the JSON for these events, see the section called "JSON for a state change event".

The channel states are the following:

- **Creating**
- **Deleting**
- **Idle**: The channel isn't running. For information about charges that you accrue when a channel is idle, see the section called "Pricing".
- **Recovering**: One or both pipelines in the channel failed, but MediaLive is restarting it.
- **Running**.
- **Starting**
- **Stopping**
- **Updating**: You changed the channel classfor a channel. This state is captured on the console but not in Amazon CloudWatch events.

States for multiplexes

MediaLive reports the state of all multiplexes.

MediaLive turns these states into CloudWatch events with the detailType set to `MediaLive Multiplex State Change`.

The multiplex states are the following:

- **Creating**
- **Deleting**
- **Idle**: The multiplex isn't running. For information about charges that you accrue when a multiplex is idle, see the section called "Pricing".
- **Recovering**: One or both pipelines in the multiplex failed, but MediaLive is restarting it.
- **Running**
- **Starting**
- **Stopping**

Alerts

MediaLive can generate alerts when a channel is running. For a list of alerts, see the section called "Alerts for channels".

You can view the alerts for each channel on the MediaLive console. For more information, see the section called "Alerts tab – Viewing alerts".

MediaLive turns alerts into CloudWatch events with the detailType set to `MediaLive Channel Alert`. For an example of the JSON for these events, see the section called "JSON for a state change event".

Metrics

For complete information about MediaLive metrics, see the section called "Monitor channels with metrics" later in this chapter.

Logs

For complete information about MediaLive logs, see the section called "CloudWatch Logs" later in this chapter.

Alerts for channels

The following table lists the alerts that MediaLive might generate for a channel. You can view these alerts in these ways:

- You can view the alerts for each channel on the MediaLive console. For more information, see the section called "Alerts tab – Viewing alerts".
- MediaLive turns alerts into CloudWatch events with the detailType set to `MediaLive Channel Alert`. For an example of the JSON for these events, see the section called "JSON for a state change event".

Alert ID	Alert wording	Description
5002	Input Image Missing	The channel was configure d with a URL to an input image (for example, an avail

Alert ID	Alert wording	Description
		blanking image). The channel can't access the file.
5007	Initial Probe is Taking Longer Than Expected	The MediaLive pipeline is not yet generating output because it is waiting for an input that it can successfully decode.
5008	Input Resource is Inaccessible	The channel configuration references a resource that MediaLive can't access. The specific resource is identified in the alert.
5010	Input Removed the Active Program	The transport stream program that was in use is no longer present in the input.
5012	SCTE-35 Input Data Could Not Be Processed	MediaLive can't process the SCTE-35 data that is being received. It's possible that the SCTE-35 PTS is not synchroni zed with the video PTS.
5101	Audio Not Detected	The channel can't decode audio in the source. Either the active input is unavailab le, or the active input doesn't contain audio, or the audio is encrypted.

Alert ID	Alert wording	Description
5102	Audio PID Missing	The audio selector for the current input specifies a PID (as the source of the audio), but that PID doesn't exist in the input.
5104	Audio Requires Dolby E Decode	The input requires Dolby E decode, but a Dolby E decode audio track selector was not specified. MediaLive might replace the audio with silence.
5201	Video Not Detected	The channel can't decode the video in the source. Either the active input is unavailab le, or the active input doesn't contain video, or the video is encrypted.
5202	Black Video Detected	Black video was detected. MediaLive might have performed an automatic input failover.
5301	HTTP Get Failed	The HTTP Get failed, so retrieval of the asset failed. Perhaps there was a network issue, or the HTTP server had a problem, or the server requires user credentials.
5302	Stopped Receiving UDP Input	A UDP input (which includes RTP, MediaConnect, and Link inputs) did not receive any packets for at least one second.

Alert ID	Alert wording	Description
5304	RTP Header Corruption	The channel is configured to receive an RTP input, but the packets received don't conform to RTP.
5305	RTMP Stream Not Found	The channel is configured to receive an RTMP input, but the specified RTMP stream is not being received.
5307	RTMP Has No Audio/Video	The channel is configured to receive an RTMP input, but the specified RTMP stream is no longer present.
5308	RTMP Disconnected	The channel is configured to receive an RTMP input, but the specified RTMP stream has disconnected.
5309	RTMP Input Connect Failed	The channel is configured to receive an RTMP input but there was a failure to connect to the RTMP URL.
5312	HTTP Request Failed When Fetching HLS Decryption Key	Don't document. This alert can't be triggered in MediaLive.
5313	HLS Segments Could Not Be Decrypted	An HLS input could not be decrypted. Check that the key provided for decryption is correct.

Alert ID	Alert wording	Description
5314	Input Double-Publishing Detected	Multiple source IP addresses are sending packets to the same MediaLive input. This situation typically causes decode errors.
5315	Data PID Missing	A transport stream data PID was specified in the channel configuration, but it isn't available in the input.
5316	Input PTS Behind PCR	A transport stream input contains video and/or audio frames that are arriving too late to decode based on comparison of their PTS (presentation timestamp) to the transport stream PCRs (program clock references). MediaLive might not be able to decode the video or audio.
5601	Input Failed Over	An input has failed and the channel is configured for automatic input failover. MediaLive has switched to the other input.
6001	ESAM HTTP Post Failed	A HTTP Post to the configure d ESAM server failed. ESAM is part of the SCTE 35 configura tion for the channel.
6002	Failed to Open UDP Socket For Write	The channel failed to open a UDP output connection.

Alert ID	Alert wording	Description
6003	Failed to Write to UDP Socket	The channel failed to write a UDP output packet.
6005	Failed to Create Output File or Socket	The channel failed to create an output file.
6006	Failed to Write to Output	The channel failed to write data to an output.
6007	Failed to Close or Finalize The Output	The channel failed to write data to an output
6008	Failed to Delete Output File	The Channel failed to delete an output file.
6010	Failed HTTP Post Output Request	An HTTP Post to an output failed.
6015	Failed to Get HTTP Output Token	The channel couldn't write to an output because it was unauthorized. For example, an HTTP access returned 401 (Unauthorized) or 403 (Forbidden).
6028	Failed to Validate Certificate Chain When Publishing	An HTTP write failed because the remote server's SSL certificate or SSH fingerprint was deemed not OK.

Alert ID	Alert wording	Description
6030	The Configured TS Muxer Bitrate is Too Low	A transport stream output was configured and the bitrate specified is not sufficient to carry the video, audio, and data that need to be carried within it. The channel includes a transport stream output. The bitrate specified in the output is too low for the combined video, audio, and data.
6031	Timecode Synchronization Threshold Exceeded	The channel was configured with a TimecodeConfigurat ion SyncThreshold, and the output timecode was resynchronized with the input timecode.
6033	Pipeline is Paused	The MediaLive pipeline has been paused.
6035	Unable to perform requested color space conversion	The channel was unable to perform the configured color space conversion.
6036	Output is Paused	One or more outputs have been paused.
6038	Nielsen Audio Watermarks could not be initialized	Nielsen audio watermarks could not be initialized.
6043	Failed To Upload Thumbnail	Video thumbnails couldn't be uploaded. MediaLive might need access to Amazon S3.

Alert ID	Alert wording	Description
6501	Large Upload Cache Backlog	The channel maintains a cache of files pending upload that it clears after successful delivery to the output. The cache has more files pending upload to the configured destination than expected, which might indicate a temporary network slowdown between MediaLive and the destination. Or it might indicate that the destination server is slower than expected.
7001	Communication Lost from Encoder	The multiplex is not receiving communication from one or more encoders.
7002	Communication Lost from Multiplex	The encoder is not receiving communication from the multiplex.
7003	Active Encoder Switched for Program	The multiplex has switched to using a different encoder pipeline for the output of a multiplex program.
7004	Active Encoder Sent Fill or Slate Frames	The active MediaLive encoder for a multiplex program inserted fill frames. Fill frames might indicate input to the encoder was lost.

Alert ID	Alert wording	Description
7005	MPTS Bitrate Overflow	More bits were generated to the multiplex then it was able to pass to its output.

Monitoring using the AWS Elemental MediaLive console

You can monitor the state and health of channels and multiplexes.

Topics

Monitoring a channel using the console

You can monitor a channel using the AWS Elemental MediaLive console to view its activity and its current state.

To monitor activity on a channel and its current state

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Channels**. (For information about the buttons on the page, see the section called "Editing a channel", *Operations: Start, stop, and pause channel*, and the section called "Creating a channel by cloning".)
3. The **Channels** page shows a list of your channels. Each line in the list provides basic information about the channel, including its state. For information about states, see the section called "States for channels".
4. To view more details about a channel, choose the name of that channel. The **Channel details** page appears.

Topics

- Destinations pane

Status tab – Viewing status information

For basic status information, look at the **Status** pane.

For information about the inputs in the channel, choose the **Details** tab.

For detailed information about the status, choose the **Health** tab. This tab provides information for the pipelines in the channel:

- Pipeline 0 and pipeline 1, if the channel is set up as a standard channel and therefore has two pipelines
- Pipeline 0, if the channel is set up as a single-pipeline channel

You can specify the period of time for the health information.

Alerts tab – Viewing alerts

MediaLive generates alerts for a channel when an issue or potential issue occurs in either pipeline in a channel. These alerts are displayed in two ways:

- On the right side of the **Status** pane, there is a count of active alerts for each pipeline.
- On the **Alerts** tab, details about each alert are displayed.

  If the alert is still active, the **Cleared** column is blank. If the alert has cleared, the column shows the timestamp for when it cleared.

For a list of MediaLive alerts, see the section called “Alerts for channels”.

Handling alerts

When an alert occurs, look at the **Alerts** tab to determine possible causes of the issue. Take steps to resolve the issue.

After you resolve the issue, MediaLive automatically clears the alert.

If you stop a channel, alerts always automatically clear.

Destinations pane

This pane has three panes:

- **Egress endpoints** – This pane shows one line for each pipeline. The **Source IP** is the channel endpoint for this pipeline. The channel endpoint is the egress from the pipeline. From this point, the content goes to the output destinations for each of the output groups in the channel.

  In a regular channel, this endpoint is in a location that MediaLive manages.

  In a channel set up for delivery via your VPC, this endpoint is in your VPC. You are responsible for ensuring that this endpoint is always available to accept the content from the channel pipeline.
- **Destinations** – This pane shows one line for each destination.

  Each output group has one destination line. Each line shows the address of the output in the one or two pipelines in the channel.
- **MediaPackage destinations** – This pane shows the channel ID that is the destination for each MediaPackage output group. The channel in MediaPackage has one or two pipelines, mapped to the one or two pipelines in MediaLive.

Monitoring a multiplex using the MediaLive Console

You can view the activity of your multiplex and its current state.

To monitor activity on a multiplex (MediaLive console)

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Multiplexes**.
3. The **Multiplexes** page shows a list of your multiplexes. Each line in the list provides basic information about the multiplex, including its state. For information about states, see the section called “States for multiplexes”.
4. To view more details about a multiplex, choose the name of that multiplex. The **Multiplex details** page appears.

Topics

- Viewing status information

Viewing status information

The **Multiplex details** page is divided into two panes. The second pane is divided into tabs.

Details tab

The **Details** tab shows the fields that you set when you created the multiplex.

It also shows this information that MediaLive assigns:

- The ARN of the multiplex.
- The ARNs of the two entitlements that MediaLive automatically creates when you create the multiplex. For more information about these entitlements, see the section called "Starting the multiplex".

Programs tab

The **Programs** tab lists the tabs that are in the multiplex. For information about programs, see the section called "Overview of multiplex and MPTS".

Bandwidth monitoring tab

The **Bandwidth monitoring** tab shows information about the bandwidth allocation for the multiplex.

To display the information as a bar chart

1. Choose **Bar chart**.
2. Choose to show the multiplex (all the programs in the multiplex) or a specific program.
3. Choose which pipeline to show.

The chart always shows the data for the most recent minute. The chart refreshes every minute.

To display the information as an area chart

1. Choose **Area chart**.
2. Set the time window. This window sets the size of the x-axis. The window always shows 60 data points. Therefore, a window of 1 hour shows a data point every minute, for example. A window of 1 day shows a data point every 24 minutes.

3. Choose to show the multiplex (all the programs in the multiplex) or a specific program.
4. Choose which pipeline to show.

Alerts tab

MediaLive generates alerts for a multiplex when an issue or potential issue occurs in either pipeline in a multiplex. These alerts are displayed in two ways:

- On the right side of the **Status** pane, there is a count of active alerts for each pipeline.
- On the **Alerts** tab, details about each alert are displayed.

If an alert is still active, the **Cleared** column is blank. If an alert has cleared, the column shows the timestamp for when it cleared.

To handle an alert

1. When an alert occurs, look at the **Alerts** tab to determine possible causes of the issue. Take steps to resolve the issue.

   After you resolve the issue, MediaLive automatically clears the alert. The **Cleared** column shows the timestamp for when it cleared.
2. If you stop a channel, alerts always automatically clear.

Tags Tab

For information about tags, see the section called “Tagging resources”.

Monitoring a channel or multiplex using Amazon CloudWatch Events

MediaLive automatically turns the following information into events in CloudWatch Events:

- Reporting on the state of a channel or multiplex.
- Alerts generated when a channel is running.

You can use Amazon CloudWatch Events to manage these events. For example, you can create event rules and deliver the events in emails or SMS messages. You can deliver events to a number

of destinations. This chapter describes how to deliver them through Amazon Simple Notification Service (SNS).

For complete information about the options for managing events using Amazon CloudWatch Events, see the CloudWatch Events User Guide.

For complete information about using Amazon SNS, see the SNS Developer Guide.

Note that events are emitted on a best-effort basis.

Topics

JSON for a state change event

Events that are based on a change of state in a channel or multiplex are identified by their `detail-type` property:

- `MediaLive Channel State Change` for a channel
- `MediaLive Multiplex State Change` for a multiplex.

Example

Following is an example of the JSON payload for a state change event. Note the `detail-type` in line 3.

```
{
    "version": "0",
    "id": "fbcbbbe3-2541-d4a3-d819-x39f522a8ce",
    "detail-type": "MediaLive Channel State Change",
    "source": "aws.medialive",
    "account": "111122223333",
    "time": "2023-03-08T18:40:59Z",
    "region": "us-west-2",
    "resources": [
```

```
        "arn:aws:medialive:us-west-2:111122223333:channel:283886"
    ],
    "detail": {
        "channel_arn": "arn:aws:medialive:us-west-2:111122223333:channel:123456",
        "state": "DELETED",
        "message": "Deleted channel",
        "pipelines_running_count": 0
    }
}
```

JSON for an alert event

Events that are based on alerts are identified by their `detail-type` property:

- `MediaLive Channel Alert` for a channel
- `MediaLive Multiplex Alert` for a multiplex.

Example

Following is an example of the JSON payload for an alert event. Note the `detail-type` in line 3.

```
{
    "version": "0",
    "id": "154769fb-9f7c-32a1-6822-26fppppe5a58",
    "detail-type": "MediaLive Channel Alert",
    "source": "aws.medialive",
    "account": "111122223333",
    "time": "2023-03-08T18:14:25Z",
    "region": "us-west-2",
    "resources": [
        "arn:aws:medialive:us-west-2:111122223333:channel:123456"
    ],
    "detail": {
        "alarm_state": "CLEARED",
        "alarm_id": "7ad616bd389832yue90aab1324bffab5b834a",
        "alert_type": "Failed to Create Output File or Socket",
        "pipeline": "0",
        "channel_arn": "arn:aws:medialive:us-west-2:111122223333:channel:123456",
        "message": "MPEGTS muxer for mediaID [1] unable to open output or stream
 [https://<path>]."
    }
```

```
}
```

Option 1: Send all MediaLive events to an email address

This option shows how to set up to send all events to a single email address. The drawback of this setup is that the email account will receive a large volume of emails. Therefore, we recommend that you don't use this setup in a production environment.

You must perform the following procedure in each Region where channels or multiplexes are running.

Step 1: Create a subscription

Create a subscription to set up a specific email address so that it automatically receives email notifications when any event occurs in MediaLive. You must identify an email recipient for the emails.

In the following procedure, we use the example of "MediaLive_alert" as the subject line and "MediaLive" as the sender of the email. We create the subscription using the Amazon Simple Notification Service (Amazon SNS) console.

To create a subscription for email notifications (Amazon SNS console)

1. Sign in to the AWS Management Console and open the Amazon SNS console at https://console.aws.amazon.com/sns/v2/home.
2. In the navigation pane, choose **Topics**, and then choose **Create new topic**.
3. In the **Create new topic** dialog box, for **Topic name**, type the name that you want for the subject line of the email, such as **`MediaLive_alert`**.
4. For **Display name**, type the name that you want for the sender of the email, such as **`MediaLive`**.
5. Choose **Create topic**.
6. Amazon SNS creates the topic and displays the ARN in the list of topics. For example, `arn:aws:sns:us-west-2:111122223333:MediaLive`, where `111122223333` is your AWS account.
7. Copy this ARN to your clipboard.
8. In the navigation pane, choose **Subscriptions**, and then choose **Create subscription**.
9. On the **Subscriptions** page, choose **Create subscription**.

10. In the **Create subscriptions** dialog box, for **Topic ARN**, type or paste the ARN.
11. For **Protocol**, choose **Email**.
12. For **Endpoint**, type the email address of the recipient. You must be able to log on to this email account because Amazon SNS sends a confirmation email to this address.
13. Choose **Create subscription**.

    Amazon SNS sends a confirmation email to the address that you specified.
14. Log on to that email account, and display the email. Choose the "Confirm subscription" link in the email to enable the subscription. A confirmation window appears in a web browser. You can close this window.

Step 2: Create a rule

You now create a rule in Amazon CloudWatch that says, "When CloudWatch receives any event from `aws.medialive`, invoke the specified SNS topic." In other words, you create a rule that sends an email to the subscribed email address.

To create a rule (Amazon CloudWatch console)

1. Sign in to the AWS Management Console and open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Events**.
3. On the **Welcome to CloudWatch Events** page, choose **Create rule**.
4. On the **Step 1** page, in **Event Source**, choose **Event Pattern**.
5. Change **Build event pattern to match** to **Custom event pattern**.
6. In the box, type the following:

```
{
  "source": [
    "aws.medialive"
  ]
}
```

7. On the pane on the right, choose **Add target**.
8. Choose **SNS topic**.
9. For **Topic**, choose the topic that you created, for example, **`MediaLive_alert`**.

10. In **Configure input**, choose **Matched event**.
11. Choose **Configure details**.
12. Type a name and optional description, and then choose **Create rule**.

Now, whenever an alert occurs in MediaLive, an event will be sent to Amazon CloudWatch. This event will trigger the rule that instructs CloudWatch to send an email to the email address that you specified in the SNS subscription.

Option 2: Send events for specific channels to an email address

You can set up a rule to send all events for one or several channels or multiplexes to one email address. You must perform this setup in each Region where channels or multiplexes are running.

Create as many subscriptions and rules combinations as you need. Follow the steps for option 1, with these differences:

- When creating the SNS subscription, you might want to add more detail to the topic, for example, **`MediaLive_notifications_channel_1234567`**.
- When creating the CloudWatch rule, you create an event pattern that identifies `aws.medialive` as the event source and the ARN for the specific channel or multiplex as the resource within that event source. For example, for a channel create this pattern:

```
{
  "source": [
    "aws.medialive"
  ],
  "resources": [
    "arn:aws:medialive:us-west-2:111122223333:channel:1234567"
  ]
}
```

The resource is the ARN for the channel or multiplex. You can obtain this ARN from the channels list or multiplexes list on the MediaLive console.

The rule for this example says, "When CloudWatch receives any event from `aws.medialive` for channel `1234567`, invoke the specified SNS topic." In other words, the rule triggers an email that is sent to the subscribed email address.

You can choose to include more than one channel or multiplex in the resources section, as shown in the following example:

```
"resources": [
  "arn:aws:medialive:us-west-2:111122223333:channel:1234567",
  "arn:aws:medialive:us-west-2:111122223333:channel:2223334"
]
```

Monitoring channels using Amazon CloudWatch metrics

You can monitor AWS Elemental MediaLive using Amazon CloudWatch metrics. CloudWatch collects raw data that it receives from MediaLive, and processes it into readable, near real-time metrics that are kept for 15 months. You use CloudWatch to view the metrics. Metrics can help you gain a better perspective about how MediaLive is performing over the short term and long term.

You can set alarms that watch for certain thresholds, and send notifications or take actions when those thresholds are met. For more information, see the Amazon CloudWatch User Guide.

Topics

Components of a metric

AWS Elemental MediaLive collects data that is the basis for metrics. It collects these *datapoints* every second and sends them immediately to Amazon CloudWatch. You can use CloudWatch to generate *metrics* for these datapoints.

A metric is a collection of datapoints that has had an aggregation (a *statistic*) applied and that has a *period* and a *time range*. For example, you can request the Dropped frames metric as an average

(the statistic) for a 1-minute period over 10 minutes (the time range). This result of this request is 10 metrics (because the range divided by the period is 10).

Statistics

MediaLive supports all the statistics offered by CloudWatch. However, some statistics aren't useful for MediaLive metrics. In the description of metrics later in this chapter, we include the recommended statistics for each metric.

Period

All MediaLive metrics have a *high resolution period*, which means that the minimum period is 1 second.

Time range

Each period has a *maximum time range*. For example, if you specify 1 day as the time range, you won't be able to retrieve metrics with a 10 second period.

Period	Maximum time range
1 second	The last 3 hours
5 seconds	
10 seconds	
30 seconds	
60 seconds	The last 360 hours (15 days)
300 seconds (5 minutes)	The last 1512 hours (63 days)
900 seconds (15 minutes)	
3600 seconds (1 hour) or longer	The last 455 days (15 months)

Periods don't have a *minimum time range*. But there is a point where the statistic you apply becomes meaningless if you have a low period. For example, assume that you set the period to

1 second. This means that CloudWatch retrieves one datapoint. You can't obtain an average, a minimum or a maximum on one datapoint. However, this doesn't mean that the metric is meaningless. Instead, the metric is for the raw datapoint, with no statistic.

Maximum storage time

Metrics are available for the last 15 months. Make sure that you specify a period that allows the time range that you want.

Dimensions for MediaLive

Each MediaLive metric includes one or two specific sets of dimensions. MediaLive metrics include the following dimensions, from the dimension with the widest scope to the dimension with the narrowest scope.

- ChannelID – Identifies a specific channel.
- Pipeline – Identifies a specific pipeline. Standard channels have two pipelines (pipeline 0 or pipeline 1). Single-pipeline channels only have pipeline 0.
- ActiveInputFailoverLabel – This dimension identifies the currently active input in a failover pair (part of the automatic input failover feature). Choose a dimension set that includes this dimension only if your channel implements automatic input failover.

  If you use this dimension, then the metric shows data only for the active input in the channel. If you don't use this dimension, the metric shows data for both inputs.
- OutputGroupName – Identifies a specific output group.
- AudioDescriptionName – Identifies a specific audio description (audio encode) among all the outputs of a channel.

Definition of a running channel

Many metrics collect data only when a channel is running.

Running means that the channel has started. It could be both ingesting and producing output. Or it could be paused, meaning that it is still ingesting but not producing output.

Keep in mind that you can view or retrieve metrics when the channel isn't running. The only requirement is that the channel has run in the last 15 months.

Pricing

There is no charge to view metrics on the **Health** tab of the MediaLiveconsole.

For information about charges to view metrics on the CloudWatch console or to retrieve metrics using a CloudWatch API, see the Amazon CloudWatch User Guide.

Viewing metrics

You can view some metrics in the MediaLive console. You can view all metrics in the CloudWatch console. You can also retrieve metrics using the CLI, the REST API, or any AWS SDK.

On the CloudWatch console, the minimum refresh rate for metrics is 30 seconds.

To view metrics on the MediaLive console

You can view some metrics in the MediaLive console. You can view those metrics for a range from the last hour up to the last week. (To view other metrics or to view historical metrics, you must use the CloudWatch console.)

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Channels**. In the **Channels** page, choose the channel you want. The **Channel details** page appears.
3. Choose the **Health** tab. The metrics that MediaLive supports on this tab appears.
4. Choose the period and time range. For example, **Past 1 day (5 min period)**.

To view metrics using the CloudWatch console

On the CloudWatch console you can view all MediaLive metrics for any range of time — the current metrics or historical metrics. There is a charge to view metrics on the CloudWatch console.

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Metrics**, then choose **All metrics**. In the bottom half of the page, the **Browse** tab shows cards with names.

   No cards appear if you are completely new to AWS, and you haven't performed an action that creates metrics in any service.
3. Select the card that is named **AWS/MediaLive**.

This card appears only if you have started at least one channel in the last 15 months in the AWS Region that is currently selected for CloudWatch. This card won't appear if have never started a MediaLive channel. In this case, come back to this procedure after you have created and started a channel.

(A card named **MediaLive** might appear in the custom namespace section of the page. This card is for the old namespace for MediaLive metrics. The two namespaces became duplicates of each other in September of 2022, so there is no advantage to choosing this card. Always choose **AWS/MediaLive**.)

4. The **Browse** tab in the bottom half of the page now shows dimensions. Choose a metric dimension. For example, choose **Channel ID**.

   The **Browse** tab now shows a table with one column that shows the chosen dimension (for example, channel IDs) and one column that shows all the metrics. You can sort the table.

5. Select one or more rows. As soon as you select a row, it appears in the graph in the top half of the page.
6. In the bottom half of the page, choose the **Graphed metrics** tab.
7. On the choices on the right of the tab, specify the **Statistic** and the **Period**.

   When you choose the period, the graph refreshes to show the maximum time range for that period. If the graph is now empty on the left, you can adjust the timeline in the choices at the top right of the graph. Choose a lower number so that the full space is filled up. For example, change **1w** to **1d**.

Alphabetical list of metrics

the section called “Active alerts”

the section called “Active outputs”

the section called “Channel input error seconds”

the section called “Dropped frames”

the section called “FEC row packets received”

the section called “FEC column packets received”

the section called “Fill msec”

the section called “Input loss seconds”

the section called “Input timecodes present”

the section called “Input video frame rate ”

the section called “Network in”

the section called “Network Out”

the section called “Output audio level dBFS”

the section called “Output audio level LKFS”

the section called “Output 4xx errors”

the section called “Output 5xx errors”

the section called “Primary input active”

the section called “RTP packets lost”

the section called “ RTP packets received”

the section called “ RTP packets recovered via FEC”

the section called “SVQ time”

Global metrics

Global metrics relate to general performance and information for AWS Elemental MediaLive.

Active alerts

The total number of alerts that are active.

Details:

- Name: ActiveAlerts
- Units: Count
- Meaning of zero: There are no active alerts

- Meaning of no datapoints: The channel isn't running
- Supported dimension sets: ChannelID, Pipeline
- Recommended statistic: Max

  All the statistics are useful for this metric.

Input metrics

Input metrics relate to the video and audio input assets that are presented to MediaLive.

Topics

FEC row packets received

The number of forward error correction (FEC) row packets received on both FEC streams (port 5002 and port 5004). A non-zero value indicates that FEC is functioning.

This metric is useful only if the channel has an RTP input that includes FEC.

Follow this guideline:

- For a channel that implements automatic input failover, we recommend that you choose the dimension set that includes the ActiveInputFailoverLabel dimension, so that you get data for only one input.

- For a channel that doesn't implement automatic input failover, don't include the ActiveInputFailoverLabel dimension set. The metric won't report any data.

Details:

- Name: FecRowPacketsReceived
- Units: Count.
- Meaning of zero: An RTP-with-FEC input was being ingested during the period but no FEC row packets were received.
- Meaning of no datapoints: There are no inputs with FEC. Or there are inputs with RTP inputs but none of those inputs are active or being prepared (by the schedule). Or you included the ActiveInputFailoverLabel in a channel that isn't set up for automatic input failover.
- Supported dimension sets:

  ChannelId, Pipeline

  ActiveInputFailoverLabel, ChannelId, Pipeline
- Recommended statistic: Sum.

FEC column packets received

The number of FEC column packets received on both FEC streams (port 5002 and port 5004). A non-zero value indicates that FEC is functioning.

This metric is useful only if the channel has an RTP input that includes FEC.

Follow this guideline:

- For a channel that implements automatic input failover, we recommend that you choose the dimension set that includes the ActiveInputFailoverLabel dimension, so that you get data for only one input.
- For a channel that doesn't implement automatic input failover, don't include the ActiveInputFailoverLabel dimension set. The metric won't report any data.

Details:

- Name: FecColumnPacketsReceived

- Units: Count.
- Meaning of zero: An RTP-with-FEC input was being ingested during the period but no FEC column packets were received.
- Meaning of no datapoints: There are no inputs with FEC. Or there are inputs with RTP inputs but none of those inputs are active or being prepared (by the schedule). Or you included the ActiveInputFailoverLabel in a channel that isn't set up for automatic input failover.
- Supported dimension sets:

  ChannelId, Pipeline

  ActiveInputFailoverLabel, ChannelId, Pipeline
- Recommended statistic: Sum.

Input timecodes present

An indicator of whether a pipeline is receiving input that includes embedded timecodes. The embedded timecode might be embedded in the source, or it might be embedded in SMPTE-2038 ancillary data. 0 (false) means it isn’t present. 1 (true) means it is present.

An embedded timecode that is inaccurate can cause problems in features that use the timecode. Therefore, it is useful to know whether the timecode that MediaLive is using is an embedded timecode or a system-clock timecode.

The timecode that is associated with the input is used in several features:

- Input clipping. This feature can use an embedded timecode or another type of timecode.
- Generating a timecode in the outputs. This feature can use an embedded timecode or another type of timecode.
- Pipeline locking. This feature works only if the input timecode is an embedded timecode; it doesn’t work with a system-clock timecode.

For detailed information about timecodes, see the section called “Timecodes”.

Follow this guideline:

- For a channel that implements automatic input failover, we recommend that you choose the dimension set that includes the ActiveInputFailoverLabel dimension, so that you get data for only one input.

- For a channel that doesn't implement automatic input failover, don't include the ActiveInputFailoverLabel dimension set. The metric won't report any data.

Details:

- Name: InputTimecodesPresent
- Units: None.
- Meaning of zero: False (there is no embedded timecode).
- Meaning of no datapoints: The channel is not running, or the channel is running but MediaLive isn’t receiving content (for example, the input is a push input and the upstream system hasn’t started pushing content). Or you included the ActiveInputFailoverLabel in a channel that isn't set up for automatic input failover.
- Supported dimension sets:

  ChannelId, Pipeline

  ActiveInputFailoverLabel, ChannelId, Pipeline
- Recommended Statistic: Minimum or maximum. The other statistics have no meaning.

Input video frame rate

The frame rate of the source video.

This metric is an indicator of the health of the input. If the value isn't stable, investigate determine if there are problems with your source,and/or problems in the network between MediaLive and the upstream system.

Follow this guideline:

- For a channel that implements automatic input failover, we recommend that you choose the dimension set that includes the ActiveInputFailoverLabel dimension, so that you get data for only one input.
- For a channel that doesn't implement automatic input failover, don't include the ActiveInputFailoverLabel dimension set. The metric won't report any data.

Details:

- Name: InputVideoFrameRate
- Units: Frames per second.
- Meaning of zero: The input has been received at some point since the channel started, but no frames were received in the current period.
- Meaning of no datapoints: No input has been received since this channel started. Or you included the ActiveInputFailoverLabel in a channel that isn't set up for automatic input failover.
- Supported dimension sets:

  ChannelID, Pipeline

  ActiveInputFailoverLabel, ChannelId, Pipeline
- Recommended statistic: Max.

Network in

The rate of traffic coming into MediaLive. This number includes all traffic received into MediaLive —push inputs, pull inputs, responses from the upstream system of a pull input, responses from the downstream system for any output, and instance traffic such as DNS resolution and NTP. Even when a channel is not ingesting, there will be some traffic.

It’s useful to set up to capture the average traffic rate over a long period. Then when you have established the normal rate, change the period to a short time, so that you can easily spot deviations from the normal rate, or collect information about how bursty a channel is.

Here are some guidelines on interpreting this metric:

- If the rate seems to be normal, then you might infer that the channel is running and successfully ingesting inputs.
- If the number is lower than normal, then your channel might be running but without inputs attached. Keep in mind that there are charges for running channels even when they aren’t ingesting input.

Details:

- Name: NetworkIn
- Units: Megabits per second.
- Meaning of zero: No traffic is being received.

- Meaning of no datapoints: The channel is not running.
- Supported dimension sets: ChannelId, Pipeline
- Recommended statistic: All the statistics are useful for this metric.

Primary input active

An indicator of whether the primary input in an automatic input failover pair is active. A value of 1 means that the primary input is active and is therefore healthy. A value of 0 means that it is inactive.

For information about input failover pairs in the automatic input failover feature, see the section called “Automatic input failover”.

This metric is useful if you have set up the automatic input failover feature with the input preference set to Primary Input Preference. The metric doesn’t provide any meaningful data if the input preference is set to Equal Input Preference.

Details:

- Name: PrimaryInputActive
- Units: None.
- Meaning of zero: False (the primary input is inactive).
- Meaning of no datapoints: The channel is not set up for automatic input failure.
- Supported dimension sets: ChannelId, Pipeline
- Recommended statistic: Minimum (primary input is inactive) or maximum (primary input is active).

RTP packets lost

The number of RTP packets that are lost in the incoming transmission. *Lost* means packets that couldn't be recovered by FEC.

This metric only applies to RTP input types.

Received packets + Recovered packets + Lost packets = Total expected for the period, if the period and dimensions for the three metrics are set identically for the three metrics.

These three RTP packet metrics are useful for monitoring the health of the input transmission. If this metric is non-zero, the first troubleshooting step is to look at the two FEC metrics, to determine whether FEC is functioning. If FEC is functioning well, the next step is to investigate problems in the upstream network.

Follow this guideline:

- For a channel that implements automatic input failover, we recommend that you choose the dimension set that includes the ActiveInputFailoverLabel dimension, so that you get data for only one input.
- For a channel that doesn't implement automatic input failover, don't include the ActiveInputFailoverLabel dimension set. The metric won't report any data.

Details:

- Name: RtpPacketsLost
- Units: Count.
- Meaning of zero: An RTP-with-FEC input was being ingested during the period, but no packets were lost.
- Meaning of no datapoints: No inputs are ingesting RTP. Or there are RTP inputs but none of those inputs are active or being prepared (by the schedule). Or you included the ActiveInputFailoverLabel in a channel that isn't set up for automatic input failover.
- Supported dimension sets:

  ChannelId, Pipeline

  ActiveInputFailoverLabel, ChannelId, Pipeline
- Recommended statistic: Sum.

RTP packets received

The number of RTP packets received in an RTP input. This number includes the main RTP source (port 5000) and the FEC data (ports 5002 and 5004).

This metric only applies to RTP input types.

Received packets + Recovered packets + Lost packets = Total expected for the period, if the periods for the three metrics are set identically.

These three RTP packet metrics are useful for monitoring the health of the input transmission.

Follow this guideline:

- For a channel that implements automatic input failover, we recommend that you choose the dimension set that includes the ActiveInputFailoverLabel dimension, so that you get data for only one input.
- For a channel that doesn't implement automatic input failover, don't include the ActiveInputFailoverLabel dimension set. The metric won't report any data.

Details:

- Name: RtpPacketsReceived
- Unit: Count.
- Meaning of zero: An RTP-with-FEC input was being ingested during the period, but no packets were received.
- Meaning of no datapoints: No inputs are ingesting RTP. Or there are inputs with RTP inputs but none of those inputs are active or being prepared (by the schedule). Or you included the ActiveInputFailoverLabel in a channel that isn't set up for automatic input failover.
- Supported dimension sets:

  ChannelId, Pipeline

  ActiveInputFailoverLabel, ChannelId, Pipeline
- Recommended statistic: Sum.

RTP packets recovered via FEC

The number of RTP packets recovered via FEC.

This metric only applies to RTP input types.

Received packets + Recovered packets + Lost packets = Total expected for the period, if the periods for the three metrics are set identically.

These three RTP packet metrics are useful for monitoring the health of the input transmission.

Follow this guideline:

- For a channel that implements automatic input failover, we recommend that you choose the dimension set that includes the ActiveInputFailoverLabel dimension, so that you get data for only one input.
- For a channel that doesn't implement automatic input failover, don't include the ActiveInputFailoverLabel dimension set. The metric won't report any data.

Details:

- Name: RtpPacketsRecoveredViaFec
- Units: Count.
- Meaning of zero: An RTP-with-FEC input was being ingested during the period, but no packets were recovered.
- Meaning of no datapoints: No inputs are ingesting RTP. Or there are inputs with RTP inputs but none of those inputs are active or being prepared (by the schedule). Or you included the ActiveInputFailoverLabel in a channel that isn't set up for automatic input failover.
- Supported dimension sets:

  ChannelId, Pipeline

  ActiveInputFailoverLabel, ChannelId, Pipeline
- Recommended statistic: Sum.

Input loss seconds

The number of seconds (the *input loss period*) for which the channel has not received packets from the source of an RTP or MediaConnect input. Each datapoint has a value between 0 and 10 seconds.

This metric is useful for monitoring the health of the input transmission.

You should look at the datapoints over several 10-second windows.

- Consistent values of 0 (all packets received) – This pattern tells you that the input is healthy.
- Consistent values of 10 (no packets received) – This pattern tells you that the input is not healthy.
- A range of values starting at 0 and ending at 0 – This pattern tells you the input was not healthy but that it has recovered. For example, 0,2,10,10,5,10,6,2,0,0,0.

- A range of values that don’t return to 0 – This pattern tells you that the input is not healthy. For example, 0,10,9,2,8,3,10,10,8,2.

Also follow this guideline:

- For a channel that implements automatic input failover, we recommend that you choose the dimension set that includes the ActiveInputFailoverLabel dimension, so that you get data for only one input.
- For a channel that doesn't implement automatic input failover, don't include the ActiveInputFailoverLabel dimension set. The metric won't report any data.

Details:

- Name: InputLossSeconds
- Units: Seconds.
- Meaning of zero: There was no input loss.
- Meaning of no datapoints: No inputs are ingesting RTP. Or there are inputs with RTP inputs but none of those inputs are active or being prepared (by the schedule). Or you included the ActiveInputFailoverLabel in a channel that isn't set up for automatic input failover.
- Supported dimension sets:

  ChannelId, Pipeline

  ActiveInputFailoverLabel, ChannelId, Pipeline
- Recommended statistic: Sum.

Channel input error seconds

The number of seconds in which the channel input contained one or more unrecoverable packets.

This metric only applies to Channel inputs of type RTP Push or MediaConnect.

This metric is useful for monitoring the health of the input. It provides a time-based measurement for packet loss.

Follow this guideline:

- For a channel that implements automatic input failover, we recommend that you choose the dimension set that includes the ActiveInputFailoverLabel dimension, so that you get data for only one input.
- For a channel that doesn't implement automatic input failover, don't include the ActiveInputFailoverLabel dimension set. The metric won't report any data.

Details:

- Name: ChannelInputErrorSeconds
- Units: Count.
- Meaning of zero: An RTP Push or MediaConnect input was being ingested and no packets were lost.
- Meaning of no datapoints: There are no RTP Push or MediaConnect inputs active or being prepared (by the schedule). Or you included the ActiveInputFailoverLabel in a channel that isn't set up for automatic input failover.
- Supported dimension sets:

  ChannelId, Pipeline

  ActiveInputFailoverLabel, ChannelId, Pipeline
- Recommended statistic: Sum.

Output metrics

Output metrics relate to the video and audio assets that have been processed by MediaLive as an output.

Topics

- Output 4xx errors
- Output 5xx errors
- SVQ time

Active outputs

The number of outputs that are being produced and successfully written to the destination.

Details:

- Name: ActiveOutputs
- Units: Count.
- Meaning of zero: None of the outputs are successfully being written to their destinations.

  If the outputs are configured to pause on input loss (according to the **Input loss action** setting for the output group), that behavior might be intentional.
- Meaning of no datapoints: The channel isn't generating output audio (it might still be starting or waiting for initial input).
- Supported dimension sets: OutputGroupName, ChannelId, Pipeline
- Recommended statistic: Minimum, which helps you identify situations when one or more outputs is not being produced.

Dropped frames

The number of input frames that MediaLive has dropped in the period. A value of 0 is expected and indicates that MediaLive is processing incoming frames in real time. A value other than 0 indicates that the encoder can't process the incoming video fast enough to keep up with real time.

Details

- Name: DroppedFrames
- Units: Count
- Meaning of zero: The encoder hasn't had to drop frames.
- Meaning of no datapoints: The channel isn't producing output. This means that it isn't running, or that it is running but it is initializing, or waiting for initial input, or paused.

- Supported dimensions sets: Pipeline, Region

- Recommended statistic: Sum.

Fill msec

The current length of time (the *fill period*) during which MediaLive has filled the video output with fill frames. The fill period starts when the pipeline does not receive content from the input within the *expected time.* The *expected time* is based on the input frame rate. The fine points of the fill frame behavior are controlled by the input loss behavior fields in the channel configuration. For information about these fields, see the section called “Global configuration – input loss behavior”.

A value of 0 means that fill frames aren't being used. A non-zero value means that fill frames are being used and that the input is unhealthy.

The count is capped at 60,000 milliseconds (1 minute), which means that after the cap, the metric will be 60,000 until it drops to zero.

Use this metric as follows:

- If you have automatic input failover enabled – This metric typically shows zero all the time, even when there is a failover. The channel fails over to the other input immediately, which means that there is no need for MediaLive to use fill frames.
- If you don’t have automatic input failover enabled – A non-zero value indicates that the input has failed, has been disrupted, or isn't keeping up with real time.

Details:

- Name: FillMsec
- Units: Count.
- Meaning of zero: The input is healthy and the output contains the expected video (rather than fill frames).
- Meaning of no datapoints: The channel isn’t producing output, which means that it isn’t running. Or that it is running but it is initializing, or waiting for initial input, or paused.
- Supported dimension sets: ChannelId, Pipeline
- Recommended statistic: Maximum, to capture the capped count when fill frames are being used.

Output audio level dBFS

The output audio level in decibels relative to full scale (dBFS).

Details:

- Name: OutputAudioLevelDbfs
- Units: Count.
- Meaning of zero: The output audio level is 0 dBFS.
- Meaning of no datapoints: The channel isn't generating output audio (it might still be starting or waiting for initial input).
- Supported dimension sets: AudioDescriptionName, ChannelId, Pipeline
- Recommended statistic: Minimum or maximum, which identify the lowest and highest audio level during the period.

Output audio level LKFS

The output audio level in loudness, K-weighted, relative to full scale (LKFS).

Details:

- Name: OutputAudioLevelLkfs
- Units: Count.
- Meaning of zero: Output audio level is 0 LFKS.
- Meaning of no datapoints: The channel isn't generating output audio (it might still be starting or waiting for initial input).
- Supported dimension sets: AudioDescriptionName, ChannelId, Pipeline
- Recommended statistic: Minimum or maximum, which identify the lowest and highest audio level during the period.

Network Out

The rate of traffic out of MediaLive. This number includes all traffic sent from MediaLive — the media output, HTTP GET requests for pull inputs, NTP traffic, and DNS traffic. Even when a channel is not delivering output, there will be some traffic.

Details:

- Name: NetworkOut
- Units: Megabits per second.
- Meaning of zero: No traffic is being sent.
- Meaning of no datapoints: The channel is not running.
- Supported dimension sets: ChannelId, Pipeline
- Recommended statistic: Average.

Output 4xx errors

The number of 4xx HTTP errors that have been received from the destination while delivering output.

Details:

- Name: Output4xxErrors
- Units: Count.
- Meaning of zero: The output is being delivered over HTTP and there are no errors.
- Meaning of no datapoints: The output is not being delivered to the destination over HTTP. Or the channel is not running.
- Supported dimension sets: OutputGroupName, ChannelId, Pipeline
- Recommended statistic: Sum.

Output 5xx errors

The number of 5xx HTTP errors that have been received from the destination while delivering output.

Details:

- Name: Output5xxErrors
- Units: Count.
- Meaning of zero: The output is being delivered over HTTP and there are no errors.
- Meaning of no datapoints: The output is not being delivered to the destination over HTTP. Or the channel is not running.

- Supported dimension sets: OutputGroupName, ChannelId, Pipeline
- Recommended statistic: Sum.

SVQ time

The percentage of time that MediaLive has had to reduce quality optimizations in order to emit output in real time. SVQ stands for speed versus quality. Any encoding task must balance emitting output in real time against a desire to produce the best quality possible. But sometimes MediaLive must reduce quality in order to encode fast enough to keep up with real time.

Details

- Name: SvqTime
- Units: Percent
- Meaning of zero: MediaLive hasn't had to reduce quality in order to produce output in real time.
- Meaning of no datapoints: The channel isn’t producing output. This means that it isn’t running, or that it is running but it is initializing, or waiting for initial input, or paused.
- Supported dimensions sets: Pipeline, Region
- Recommended statistic: Max.

Pipeline locking metrics

Pipeline locking metrics relate to the synchronization of MediaLive pipelines.

Topics

Pipelines locked

An indicator of whether the two pipelines are synchronized with each other. This metric applies only to standard channels and only to HLS, MediaPackage, Microsoft Smooth, and UDP outputs in that channel. MediaLive uses pipeline locking to ensure that the two pipelines are synchronized with each other.

With this metric, you must determine whether the channel you are looking at is a standard channel and has at least one eligible output. If this scenario applies, then a value of 1 means that all the eligible pairs of pipelines are synchronized. A value of 0 means that at least one pair of eligible pipelines is not synchronized.

For any other scenario, the metric is always 0. For example, if the channel is a standard channel with no eligible outputs. Or the channel isn't a standard channel.

Details:

- Name: PipelinesLocked
- Units: Not applicable.
- Meaning of zero: False (the eligible pipelines are not synchronized), but only if the channel is standard.
- Meaning of no data points: The channel is not running.
- Supported dimension sets: ChannelId, Pipeline
- Recommended statistic: Minimum (Value is 0).

Monitoring a channel using Amazon CloudWatch Logs

MediaLive produces channel logs that contain detailed information about activity in a channel. The logs provide a sequential description of activity that occurs in the channel. These logs can be useful when the information in alerts (the section called "Monitor with CloudWatch events") does not provide enough information to resolve an issue on the channel.

Topics

About channel logs

MediaLive produces channel logs that contain detailed information about activity in a channel. The logs provide a sequential description of activity that occurs in the channel. These logs can be useful when the information in alerts (the section called "Monitor with CloudWatch events") does not provide enough information to resolve an issue on the channel.

There are two sets of channel logs:

- Channel encoder logs. You must enable these logs.
- Channel as-run logs. MediaLive always produces these logs.

Comparison of types of logs

Features that are the same in both types of logs

Both types of logs are sent to Amazon CloudWatch Logs. You can use the standard features of CloudWatch Logs to view and manage the logs. For more information, see Amazon CloudWatch Logs User Guide.

Features that are different in the two types of logs

The following table describes the differences between channel encoder logs and channel as-run logs.

	Encoder logs	As-run logs
Trigger for creation	You must enable these logs in order for MediaLive to produce them.	MediaLive always produces these logs.
Level of detail	You can set a logging level to control the detail collected.	You can't change the logging level.
Cost	There is a cost for these logs, as part of your charges for Amazon CloudWatch Logs. See Amazon CloudWatch Pricing. Remember to remove the logs after you delete the channel.	These logs are free.
CloudWatch log stream	The log stream is named after the ARN/pipeline.	The log stream is named after the ARN/pipeline with

	Encoder logs	As-run logs
		`_as_run` appended to the name.
Automation	You should not automate any processing based on the wording in these logs because that wording is subject to change. (By comparison, you can automate based on the wording in alerts, which are accessed using CloudWatch Events, because the wording of alerts does not change.)	You can automate based on the wording in these logs.

Enabling channel encoder logs

You enable channel encoder logs for an individual channel on the MediaLive console. You enable logging and set the logging level (error, warning, info, or debug) on a per channel basis. The channel must be idle in order to enable or disable logging.

You don't need to enable as-run logs. MediaLive always produces these logs.

To enable a channel encoder log (MediaLive console)

1. If you are a returning user of MediaLive, check with your administrator that your deployment has been set up in AWS IAM to support channel logs.
2. Your administrator might instruct you to update the `MediaLiveAccessRole` permission in one of the channels. If you are given this instruction, you must edit a channel (choose any idle channel), display the **Channel and input details** page, and choose the **Update** button. When the role is updated in one channel, the change applies to all channels.
3. To enable encoder logs in a new channel, set up logging during creation.

   To enable encoder logs in an existing channel, edit the channel; this channel must be idle.

In both cases, on the **General settings** page, in the **Channel logging** section, choose **Logging**. Choose a level other than **DISABLED**. For more information, see the section called “Logging”.

4. You or an administrator can also go into CloudWatch Logs and set an expiry date for the logs.

Disabling channel encoder logs

You disable the capture of encoder-related logging information for an individual channel on the MediaLive console. Edit the channel, and on the **General settings** page, on the **Channel logging** section, choose **Logging**. Set the level to **DISABLED**.

Working with logs

You view both encoder logs and as-run logs on the CloudWatch Logs console, in the same way that you view logs for any service.

You don't have to set up the logs, logging groups, or log streams on the CloudWatch Logs console because MediaLive automatically sets them up for you.

- Log group – The log group is always the following: **ElementalMediaLive**.
- Log stream –The log stream is named as follows:
  - Encoder logs – named after the ARN/pipeline.
  - As-run logs – named after the ARN/pipeline with `_as_run` appended.

  For example:

  `arn_aws_medialive_us-west-2_111122223333_channel_5106412_0`

  `arn_aws_medialive_us-west-2_111122223333_channel_5106412_0_as_run`

  Where `5106412` is the channel ID and `0` is the pipeline.

Content of encoder logs

The logs are in JSON format:

```
{
    "encoder_pipeline": 0,
    "severity": "I",
```

```
    "timestamp": "2018-05-21T16:36:41.650318",
    "channel_arn": "arn:aws:medialive:us-west-2:111122223333:channel:5106412",
    "logger_name": "",
    "message": "Probing input media..."
  },
.
.
.
]
```

The data is the following:

- `encoder_pipeline`: `0` or `1` (if the channel is set up as a standard channel and therefore has two pipelines).
- `severity`: A letter. The logging level (which you set when you enable logging) controls which severities could appear in logs. For more information, see Log Levels and Verbosities.
- `timestamp`: The time in ISO 8601 format: yyyy - mm - dd T hh : mm : ss : decimal fraction of second.
- `channel_arn`: The ARN plus the channel ID. In the preceding example, the channel has ID `5106412`.
- `logger_name`: This might be blank or might specify a name that ties a series of related messages together.
- `message`: The message. Remember that the wording is subject to change, so you should not automate against it.

Log levels and verbosities for encoder logs

To use this table, find a level in the first column then read across to identify the message severities that will appear in the logs with this logging level.

Level	Debug messages	Info messages	Warning messages	Critical messages	Fatal messages
DEBUG	Yes	Yes	Yes	Yes	Yes
INFO		Yes	Yes	Yes	Yes
WARNING			Yes	Yes	Yes

Level	Debug messages	Info messages	Warning messages	Critical messages	Fatal messages
ERROR				Yes	Yes

Managing log storage

When you delete a channel, the associated logs remain in CloudWatch Logs. You will continue to be charged for their storage until you delete them. To delete logs, change the log data retention. All the data that is older than the retention setting that you specify will be deleted. For more information, see Amazon CloudWatch Logs User Guide. The **Log group** for the logs is **ElementalMediaLive**.

Logging MediaLive API calls with AWS CloudTrail

AWS Elemental MediaLive is integrated with AWS CloudTrail, CloudTrail is service that provides a record of actions taken by a user, role, or an AWS service. CloudTrail captures all API calls for MediaLive as events. The calls captured include calls from the MediaLive console and code calls to the MediaLive API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for MediaLive. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to MediaLive, the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, see the AWS CloudTrail User Guide.

MediaLive information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in MediaLive, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing Events with CloudTrail Event History.

For an ongoing record of events in your AWS account, including events for MediaLive, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally,

you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- Overview for Creating a Trail
- CloudTrail Supported Services and Integrations
- Configuring Amazon SNS Notifications for CloudTrail
- Receiving CloudTrail Log Files from Multiple Regions and Receiving CloudTrail Log Files from Multiple Accounts

All MediaLive actions are logged by CloudTrail and are documented in the https://docs.aws.amazon.com/medialive/latest/apireference/.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the CloudTrail userIdentity Element.

Understanding MediaLive log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following example shows a CloudTrail log entry. The example shows the entry for one API call. The call is made by the identity that is specified in `userIdentity`, in this case a user with the user name `santosp`. The call was a `CreateInput` operation coming from the AWS CLI (as specified in `userAgent`) running on a computer with the IP address 203.0.113.33:

```
 {
“eventVersion”: “1.05",
```

```
    “userIdentity”: {
        “type”: “IAMUser”,
        “principalId”: “AIDACKCEVSQ6C2EXAMPLE”,
        “arn”: “arn:aws:iam::111122223333:user/santosp”,
        “accountId”: “111122223333”,
        “accessKeyId”: “AKIAOSFODNN7EXAMPLE”,
        “userName”: “santosp”
    },
    “eventTime”: “2019-01-17T21:21:17Z”,
    “eventSource”: “medialive.amazonaws.com”,
    “eventName”: “CreateInput”,
    “awsRegion”: “us-west-2”,
    “sourceIPAddress”: “203.0.113.33”,
    “userAgent”: “aws-cli/1.16.86 Python/2.7.15 Darwin/17.7.0 botocore/1.12.76”,
    “requestParameters”: {
        “mediaConnectFlows”: [],
        “inputSecurityGroups”: [
            “9999999”
        ],
        “sources”: [],
        “roleArn”: “MediaLiveAccessRole”,
        “requestId”: “1111aaaa-9604-4459-a160-46a28ae166",
        “name”: “live-studio-feed”,
        “type”: “RTP_PUSH”,
        }
    },
    “responseElements”: {
        “input”: {
            “arn”: “arn:aws:medialive:us-west-2:111122223333:input:7780651”,
            “id”: “7780651”,
            “name”: “live-studio-feed”,
            “type”: “RTP_PUSH”,
            “sources”: [],
            “destinations”: [
                {
                    “url”: “rtp://198.51.100.10:1935”,
                    “ip”: “198.51.100.10:1935”,
                    “port”: “1935”
                },
                {
                    “url”: “rtp://192.0.2.131:1935”,
                    “ip”: “192.0.2.131:1935”,
                    “port”: “1935”
                }
```

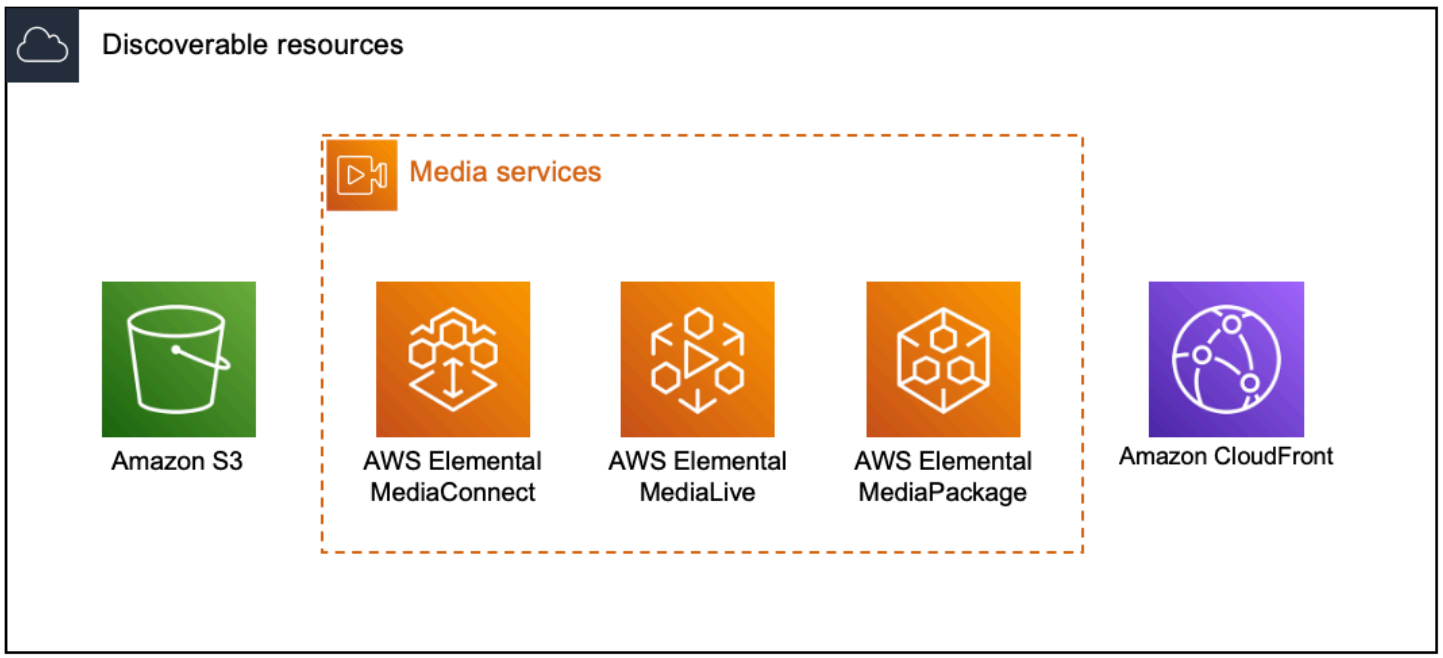
```
            ],
            “mediaConnectFlows”: [],
            “state”: “DETACHED”,
            “attachedChannels”: [],
            “securityGroups”: [
                “9999999”
            ],
            “roleArn”: “”
        }
    },
    “requestID”: “d2f882ac-1a9d-11e9-a0e5-afe6a8c88993”,
    “eventID”: “ebbe0290-7a1b-4053-a219-367404e0fe96”,
    “readOnly”: false,
    “eventType”: “AwsApiCall”,
    “recipientAccountId”: “111122223333"
}
```

Monitoring AWS media services with workflow monitor

Workflow monitor is a tool for the discovery, visualization, and monitoring of AWS media workflows. Workflow monitor is available in the AWS console and API. You can use workflow monitor to discover and create visual mappings of your workflow's resources, called *signal maps*. You can create and manage Amazon CloudWatch alarm and Amazon EventBridge rule templates to monitor the mapped resources. The monitoring templates you create are transformed into deployable AWS CloudFormation templates to allow repeatability. AWS-recommended alarm templates provide predefined best-practice monitoring.

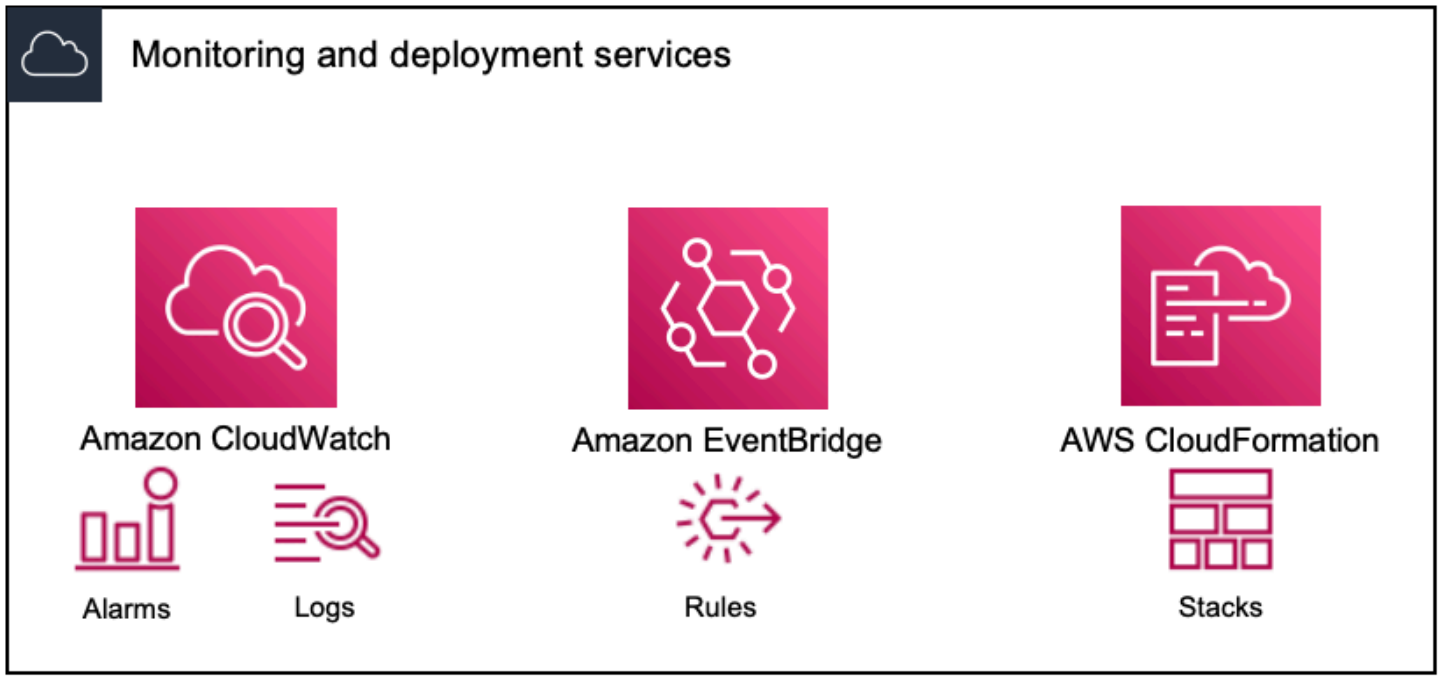
Discover

Utilize signal maps to automatically discover interconnected AWS resources associated with your media workflow. Discovery can begin at any supported service resource and creates an end-to-end mapping of the workflow. Signal maps can be used as stand-alone visualization tools or enhanced with monitoring templates.



Monitor

You can create custom CloudWatch alarm and EventBridge rule templates to monitor the health and status of your media workflows. Best practice alarm templates are available to import into your workflow monitor environment. You can use the best practice alarm templates as they are, or edit them to better fit your workflow. Any templates you create are transformed into AWS CloudFormation templates for repeatable deployment.



Note

There is no direct cost for using workflow monitor. However, there are costs associated with the resources created and used to monitor your workflow.
When monitoring is deployed, Amazon CloudWatch and Amazon EventBridge resources are created. When using the AWS Management Console, prior to deploying monitoring to a signal map, you will be notified of how many resources will be created. For more information about pricing, see: CloudWatch pricing and EventBridge pricing.
Workflow monitor uses AWS CloudFormation templates to deploy the CloudWatch and EventBridge resources. These templates are stored in a standard class Amazon Simple Storage Service bucket that is created on your behalf, by workflow monitor, during the deployment process and will incur object storage and recall charges. For more information about pricing, see: Amazon S3 pricing.
Previews generated in the workflow monitor signal map for AWS Elemental MediaPackage channels are delivered from the MediaPackage Origin Endpoint and will incur Data Transfer Out charges. For pricing, see: MediaPackage pricing.

Components of workflow monitor

Workflow monitor has four major components:

- CloudWatch alarm templates - Define the conditions you would like to monitor using CloudWatch. You can create your own alarm templates, or import predefined templates created by AWS. For more information, see: CloudWatch alarm groups and templates
- EventBridge rule templates - Define how EventBridge sends notifications when an alarm is triggered. For more information, see: EventBridge rule groups and templates
- Signal maps - Use an automated process to create AWS Elemental workflow maps using existing AWS resources. The signal maps can be used to discover resources in your workflow and deploy monitoring to those resources. For more information, see: Workflow monitor signal maps
- Overview - The overview page allows you to directly monitor the status of multiple signal maps from one location. Review metrics, logs, and alarms for your workflows. For more information, see: Workflow monitor overview

Supported services

Workflow monitor supports automatic discovery and signal mapping of resources associated with the following services:

- AWS Elemental MediaLive
- AWS Elemental MediaPackage
- AWS Elemental MediaConnect
- Amazon S3
- Amazon CloudFront

Topics

Configuring workflow monitor

To setup workflow monitor for the first time; you create the alarm and event templates, and discover signal maps that are used to monitor your media workflows. The following guide contains the steps necessary to setup both Administrator and Operator level IAM roles, create workflow monitor resources, and deploy monitoring to your workflows.

Topics

Getting started with workflow monitor

The following steps provide a basic overview of using workflow monitor for the first time.

1. Setup workflow monitor IAM permissions for administrator and operator level roles: Workflow monitor IAM policies
2. Build alarm templates or import predefined templates created by AWS: CloudWatch alarms

3. Build notification events that will be delivered by EventBridge: EventBridge rules
4. Discover signal maps using your existing AWS Elemental resources: Signal maps
5. Attach the alarm templates and notification rules to your signal map: Attach templates
6. Deploy the templates to begin monitoring the signal map: Deploy monitoring
7. Monitor and review your workflow monitor resources using the overview section of the AWS console: Overview

Workflow monitor IAM policies

Workflow monitor interacts with multiple AWS services to create signal maps, build CloudWatch and EventBridge resources, and AWS CloudFormation templates. Because workflow monitor interacts with a wide range of services, specific AWS Identity and Access Management (IAM) policies must be assigned for these services. The following examples indicate the necessary IAM policies for both administrator and operator IAM roles.

Administrator IAM policy

The following example policy is for an administrator-level workflow monitor IAM policy. This role allows for the creation and management of workflow monitor resources and the supported service resources that interact with workflow monitor.

```
                    {
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "cloudwatch:List*",
      "cloudwatch:Describe*",
      "cloudwatch:Get*",
      "cloudwatch:PutAnomalyDetector",
      "cloudwatch:PutMetricData",
```

```
        "cloudwatch:PutMetricAlarm",
        "cloudwatch:PutCompositeAlarm",
        "cloudwatch:PutDashboard",
        "cloudwatch:DeleteAlarms",
        "cloudwatch:DeleteAnomalyDetector",
        "cloudwatch:DeleteDashboards",
        "cloudwatch:TagResource",
        "cloudwatch:UntagResource"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:List*",
        "cloudformation:Describe*",
        "cloudformation:CreateStack",
        "cloudformation:UpdateStack",
        "cloudformation:DeleteStack",
        "cloudformation:TagResource",
        "cloudformation:UntagResource"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudfront:List*",
        "cloudfront:Get*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeNetworkInterfaces"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "events:List*",
        "events:Describe*",
```

```
        "events:CreateEventBus",
        "events:PutRule",
        "events:PutTargets",
        "events:EnableRule",
        "events:DisableRule",
        "events:DeleteRule",
        "events:RemoveTargets",
        "events:TagResource",
        "events:UntagResource"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:Describe*",
        "logs:Get*",
        "logs:TagLogGroup",
        "logs:TagResource",
        "logs:UntagLogGroup",
        "logs:UntagResource"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "mediaconnect:List*",
        "mediaconnect:Describe*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "medialive:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "mediapackage:List*",
        "mediapackage:Describe*"
```

```
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "mediapackagev2:List*",
        "mediapackagev2:Get*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "mediapackage-vod:List*",
        "mediapackage-vod:Describe*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "mediatailor:List*",
        "mediatailor:Describe*",
        "mediatailor:Get*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "resource-groups:ListGroups",
        "resource-groups:GetGroup",
        "resource-groups:GetTags",
        "resource-groups:GetGroupQuery",
        "resource-groups:GetGroupConfiguration",
        "resource-groups:CreateGroup",
        "resource-groups:UngroupResources",
        "resource-groups:GroupResources",
        "resource-groups:DeleteGroup",
        "resource-groups:UpdateGroupQuery",
        "resource-groups:UpdateGroup",
        "resource-groups:Tag",
        "resource-groups:Untag"
```

```
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:*"
      ],
      "Resource": "arn:aws:s3:::workflow-monitor-templates*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "sns:TagResource",
        "sns:UntagResource"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "tag:Get*",
        "tag:Describe*",
        "tag:TagResources",
        "tag:UntagResources"
      ],
      "Resource": "*"
    }
  ]
}
```

Operator IAM policy

The following example policy is for an operator-level workflow monitor IAM policy. This role allows for limited and read-only access to the workflow monitor resources and the supported service resources that interact with workflow monitor.

```
                    {
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:List*",
        "cloudwatch:Describe*",
        "cloudwatch:Get*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:List*",
        "cloudformation:Describe*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudfront:List*",
        "cloudfront:Get*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeNetworkInterfaces"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "events:List*",
        "events:Describe*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
```

```
        "logs:Describe*",
        "logs:Get*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "mediaconnect:List*",
        "mediaconnect:Describe*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "medialive:List*",
        "medialive:Get*",
        "medialive:Describe*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "mediapackage:List*",
        "mediapackage:Describe*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "mediapackagev2:List*",
        "mediapackagev2:Get*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "mediapackage-vod:List*",
        "mediapackage-vod:Describe*"
      ],
```

```
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "mediatailor:List*",
        "mediatailor:Describe*",
        "mediatailor:Get*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Resource": "arn:aws:s3:::workflow-monitor-templates*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "tag:Get*",
        "tag:Describe*"
      ],
      "Resource": "*"
    }
  ]
}
```

Workflow monitor groups and templates

Before you can deploy workflow monitoring to a signal map, you must create the groups and templates for CloudWatch alarms and EventBridge notifications. The CloudWatch templates define what scenarios and thresholds will be used to trigger the alarms. The EventBridge templates will determine how these alarms are reported to you.

If you only want mappings of your connected resources and do not want to use the monitoring template capabilities of workflow monitor, signal maps can be used without CloudWatch and EventBridge templates. For more information about using signal maps, see: Signal maps

Topics

CloudWatch alarm groups and templates

Workflow monitor alarms allow you to use existing CloudWatch metrics as the foundation of alarms for your signal maps. You can create an alarm template group to sort and classify the types of alarming that is important to your workflow. Within each alarm template group, you create alarm templates with specific CloudWatch metrics and parameters that you want to monitor. You can create your own alarm templates or import recommended alarm templates created by AWS. After creating an alarm template group and alarm templates within that group, you can attach one or more of these alarm template groups to a signal map.

You must create an alarm template group first. After you have created an alarm template group, you can create your own templates or use recommended templates created by AWS. If you want to create your own alarm templates, continue on this page. For more information about importing recommended templates, see: Recommended templates

This section covers the creation of CloudWatch alarms using workflow monitor. For more information about how the CloudWatch service handles alarms and details of the alarm components, see: Using CloudWatch alarms in the *Amazon CloudWatch User Guide*

Creating alarm template groups

To create an alarm template group

1. From the workflow monitor console's navigation pane, select **CloudWatch alarm templates**.
2. Select **Create alarm template group**.
3. Give the alarm template group a unique **Group name** and optional **Description**.
4. Select **Create**, You will be taken to the newly created alarm template group's details page.

Creating alarm templates

To create an alarm template

1. From the alarm template group's details page, select **Create alarm template**.
2. Give the alarm template a unique **Template name** and optional **Description**.

3. In the **Choose metric** section:

    1. Select a **Target Resource Type**. The target resource type is a resource for the respective service, such as a channel for MediaLive and MediaPackage or a flow for MediaConnect.
    2. Select a **Metric Name**. This is the CloudWatch metric that acts as the foundation for the alarm. The list of metrics will change depending on the selected **Target Resource Type**.

4. In the **Alarm settings** section:

    > **Note**
    >
    > For more information about how the CloudWatch service handles alarms and details of the alarm components, see: Using CloudWatch alarms in the *Amazon CloudWatch User Guide*

    1. Select the **Statistic**. This is a value such as a **Sum** or an **Average** that will be used to monitor the metric.
    2. Select the **Comparison Operator**. This field references the **Threshold** that you set in the next step.
    3. Set a **Threshold**. This is a numeric value that the **Comparison Operator** uses to determine greater than, less than, or equal to status.
    4. Set a **Period**. This is a time value, in seconds. The **Period** is the length of time that the **Statistic**, **Comparison Operator**, and **Threshold** interact to determine if the alarm gets triggered.
    5. Set the **Datapoints**. This value determines how many datapoints are needed to trigger the alarm.
    6. Select how to **Treat Missing Data**. This selection determines how this alarm reacts to missing data.

5. Select **Create** to complete the process.

An example of a completed alarm template could have the following parameters: A MediaConnect flow **Target Resource Type** is monitored for the Disconnections **Metric Name**. The **Statistic** value is set to Sum with a **Comparison Operator** of "greater than or equal to" and a **Threshold** of 10. The **Period** is set to 60 seconds, and only requires 1 out of 1 **Datapoints**. **Treat Missing Data** is set to "ignore."

The result of these settings is: workflow monitor will monitor for disconnections on the flow. If 10 or more disconnections occur within 60 seconds, the alarm will be triggered. 10 or more disconnections in 60 seconds only needs to happen one time for the alarm to be triggered.

Recommended alarm templates

Workflow monitor's recommended templates are a curated selection of AWS Elemental service metrics with predefined alarm settings appropriate for the metric. If you do not want to create customized alarm templates, recommended templates provide you with best-practice monitoring templates that are created by AWS.

Workflow monitor contains recommended template groups for each supported service. These groups are designed to apply best-practice monitoring to specific types of workflows. Each template group contains a curated selection of alarms configured from service-specific metrics. For example, a recommended template group for a MediaLive multiplex workflow will have a different set of preconfigured metrics than a MediaConnect CDI workflow.

To use recommended alarm templates

1. Follow the steps to create an alarm template group, or select an existing one.
2. In the **Alarm templates** section, select **Import**. You will need to import the AWS recommended templates into your template group.
3. Use the **CloudWatch alarm template groups** dropdown to select an AWS recommended group. These groups contain curated alarms for specific services.
4. Select the templates to import using the check boxes. Each template will list its metrics, preconfigured monitoring values, and provide a description of the metric. When you are done selecting templates, select the **Add** button.
5. The selected templates will move to the **Alarm template(s) to import** section. Review your choices and select **Import**.
6. After the import is complete, the selected templates will be added to the template group. If you want to add more templates, repeat the import process.
7. Imported templates can be customized after import. Alarm settings can be modified to fit your alarming needs.

EventBridge rule groups and templates

CloudWatch uses Amazon EventBridge rules to send notifications. You can send notifications based on event templates you create. You begin by creating an event template group. In that event template group, you create event templates that determine what conditions create a notification and who is notified.

This section covers the creation of EventBridge rules using workflow monitor. For more information about how the EventBridge service uses rules, see: EventBridge rules in the *Amazon EventBridge User Guide*

Creating event template groups

To create an event template group

1. From the workflow monitor console's navigation pane, select **EventBridge rule templates**.
2. Select **Create event template group**.
3. Give the alarm template group a unique **Group name** and optional **Description**.
4. Select **Create**, You will be taken to the newly created alarm template group's details page.

Creating event templates

To create an event template

1. From the event template group's details page, select **Create event template**.
2. Give the event template a unique **Template name** and optional **Description**.
3. In the **Rule settings** section:
    1. Select an **Event type**. When selecting an event type, you can choose between several events created by AWS or select **Signal map active alarm** to use an alarm created by an alarm template.
    2. Select a **Target service**. This determines how you would like to be notified of this event. You can select Amazon Simple Notification Service or CloudWatch logs.
    3. After selecting a target service, select a **Target**. This will be a Amazon SNS topic or a CloudWatch log group, depending on your target service selection.
4. Select **Create** to complete the process.

Workflow monitor signal maps

Signal maps are visual mappings of AWS resources in your media workflow. You can use workflow monitor to start the signal map discovery on any of the supported resource types. During the discovery process, workflow monitor will automatically and recursively map all connected AWS resources. After the signal map has been created, you can use the workflow monitor console to do things like deploy monitoring templates, view metrics, and view details of the mapped resources.

Topics

Creating signal maps

To create a signal map

1. From the workflow monitor console's navigation pane, select **Signal maps**.
2. Select **Create signal map**.
3. Give the signal map a **Name** and **Description**.
4. In the **Discover new signal map** section, resources in the current account and selected region are displayed. Select a resource to begin signal map discovery. The selected resource will be the starting point for discovery.
5. Select **Create**. Allow a few moments for the discovery process to complete. After the process is complete, you will be presented with the new signal map.

> **Note**
>
> Previews generated in the workflow monitor signal map for AWS Elemental MediaPackage channels are delivered from the MediaPackage Origin Endpoint and will incur Data Transfer Out charges. For pricing, see: MediaPackage pricing.

Viewing signal maps

Signal map views

After selecting a signal map, you have two views that can be used to monitor or configure the signal map. **Monitor signal map** and **Configure signal map** is a context-sensitive button found in the upper-right of the signal map console section.

If you select the signal map using the **Signal maps** section of the navigation pane, your signal map will be displayed in the configuration view. The configuration view allows you to make changes to the template groups attached to this signal map, deploy the attached templates, and view the basic details and tags of the signal map.

If you select the signal map using the **Overview** section of the navigation pane, your signal map will be displayed in monitoring view. The monitoring view displays the CloudWatch alarms, EventBridge rules, alerts, logs, and metrics for this signal map.

The view can be changed at any time by selecting the **Monitor/Configure signal map** button in the upper-right. The configuration view requires administrator-level IAM permissions. Required IAM permissions can be viewed here: Workflow monitor IAM policies

Navigating the signal map

A signal map will contain nodes for every supported AWS resource discovered by workflow monitor. Certain resources, such as MediaLive channels and MediaPackage endpoints can display thumbnail previews of the content, if thumbnail previews are available.

Selecting a resource node, and selecting **View selected resource details** from the **Actions** dropdown menu will take you to the associated service's details page. For example, selecting a MediaLive channel and selecting **View selected resource details** will open the MediaLive console's details page for that channel.

Selecting a resource node will filter the list of active alarms to only that node. If you select the resource's **Target ARN** in the active alarm, you will be taken to the associated service's details page, with the selected resource open.

Attaching alarm and event templates to your signal map

After you have created alarm and event templates, you need to attach these to a signal map. Any of the alarm and event templates you have created can be attached to any discovered signal maps.

To attach alarm and event templates to your signal map

1. From the workflow monitor console's navigation pane, select **Signal maps** and select the signal map you want to work with.
2. In the upper-right of the signal map page, in the **CloudWatch alarm template groups** tab, select **Attach CloudWatch alarm template groups**.
    1. In the new section that opens, choose all of the alarm template groups that you want to apply to this signal map, then select **Add**. This will cause the selected alarm template groups to move to the **Attached CloudWatch alarm template groups** section.
    2. Selecting **Save** will save your changes and return you to the signal map page.
3. At the right of the signal map page, select the **EventBridge rule template groups** tab then select **Attach EventBridge rule template groups**.
    1. In the new section that opens, choose all of the event template groups that you want to apply to this signal map, then select **Add**. This will cause the selected rule template groups to move to the **Attached EventBridge rule template groups** section.
    2. Selecting **Save** will save your changes and return you to the signal map page.
4. You have assigned CloudWatch alarm and EventBridge rule templates to the signal map, but the monitoring is not yet deployed. The next section will cover the deployment of the monitoring resources.

Deploying templates to your signal map

After you have attached the alarm and event templates to your signal map, you must deploy the monitoring. Until the deployment is complete, the monitoring of your signal map will not be active.

Workflow monitor will only deploy alarms that are relevant to the selected signal map. For example, the attached alarm template group might contain alarms for multiple services, such as MediaLive, MediaPackage, and MediaConnect. If the selected signal map only contains MediaLive resources, no MediaPackage or MediaConnect alarms will be deployed.

To deploy the monitoring templates

1. After attaching alarm and event template groups to your signal map and saving your changes, select **Deploy monitor** in the **Actions** dropdown menu.

2. You will be asked to confirm the deployment and presented with the number of CloudWatch and EventBridge resources that will be created. If you would like to proceed, select **Deploy**.

   > **Note**
   >
   > There is no direct cost for using workflow monitor. However, there are costs associated with the resources created and used to monitor your workflow.
   > When monitoring is deployed, Amazon CloudWatch and Amazon EventBridge resources are created. When using the AWS Management Console, prior to deploying monitoring to a signal map, you will be notified of how many resources will be created. For more information about pricing, see: CloudWatch pricing and EventBridge pricing.
   > Workflow monitor uses AWS CloudFormation templates to deploy the CloudWatch and EventBridge resources. These templates are stored in a standard class Amazon Simple Storage Service bucket that is created on your behalf, by workflow monitor, during the deployment process and will incur object storage and recall charges. For more information about pricing, see: Amazon S3 pricing.

3. The status of the deployment is displayed next to the name of the signal map. The deployment status is also visible in the **Stacks** section of the AWS CloudFormation console. After a few moments of resource creation and deployment, your signal map monitoring will begin.

Updating signal maps and underlying resources

If a change is made to your workflow, you might need to rediscover the signal map and redeploy monitoring resources. Workflow monitor is a visualization and monitoring tool that does not have the ability to make any changes to your workflow. Signal maps represent a point-in-time visualization of your workflow. In the event that you add, remove, or significantly modify parts of your media workflow, we recommend that you rediscover the signal map. If you have monitoring resources attached to the signal map, we recommend you redeploy monitoring after the rediscovery process.

To rediscover a signal map

1. From the workflow monitor console's navigation pane, select **Signal maps** and select the signal map you want to work with.
2. Verify that you are in the **Configure signal map** view. For more information about changing views, see: View signal maps

3. In the upper-right of the signal map page, select the **Actions** dropdown menu. Select **Rediscover**.
4. You will be presented with the rediscovery screen. Select a resource that is a part of the workflow you are rediscovering. Select the **Rediscover** button.
5. The signal map will be rebuilt according to the current workflow. If you need to redeploy monitoring resources, stay on this signal map's page. Any previously attached monitoring templates will remain attached, but will need to be redeployed.

To redeploy monitoring templates after a signal map rediscovery

1. After the rediscovery, you will be directed to the updated signal map. To redeploy the monitoring templates, select **Deploy monitor** from the **Actions** dropdown menu.
2. You will be asked to confirm the deployment and presented with the number of any CloudWatch and EventBridge resources that will be created. If you would like to proceed, select **Deploy**.
3. The status of the deployment is displayed next to the name of the signal map. After a few moments of resource creation and deployment, your signal map monitoring will begin.

Deleting signal maps

If you not longer need a signal map, it can be deleted. If you have monitoring templates deployed on the signal map, the deletion process will ask you to delete any CloudWatch and EventBridge resources that have been deployed to this signal map. Deleting the deployed resources does not affect the templates that created them. This resource deletion is to ensure that you do not have CloudWatch and EventBridge resources that are deployed but not used.

To delete a signal map

1. From the workflow monitor console's navigation pane, select **Signal maps** and select the radio button next to the signal map you want to delete.
2. Select the **Delete** button. You will be asked to confirm the deletion of the monitoring resources. Select **Delete** to begin the monitoring resource deletion process.
3. The **Monitor deployment** column will display the current status. When the status has changed to **DELETE_COMPLETE**, select the **Delete** button again.

4. You will be asked to confirm deletion of the signal map. Select **Delete** to proceed and delete the signal map.

Workflow monitor quotas

The following section contains quota for workflow monitor resources. Each quota is on a "per account" basis. You cannot exceed the following quotas on a single AWS account. These quotas cannot be increased.

Quotas

Resource type	Quota
CloudWatch alarm template groups	20
CloudWatch alarm templates	200
EventBridge rule template groups	20
EventBridge rule templates	200
Signal maps	30
Signal maps: resource nodes in a single signal map	30
Signal maps: CloudWatch alarm template groups attached to a single signal map	5
Signal maps: EventBridge rule template groups attached to a single signal map	5

Using workflow monitor

Use the **overview** and **signal maps** sections of the workflow monitor console to review the current status of the workflows and any associated alarms, metrics, and logs.

Topics

- Workflow monitor overview

Workflow monitor overview

The **Overview** section of the workflow monitor console is a dashboard that provides at-a-glance information about your signal maps. In the overview section, you can see the current state of each signal map's monitoring, as well as CloudWatch metrics and any associated CloudWatch logs. You can select any signal map to be taken to that signal maps console page.

Overview filtering

Using the **Search** bar in the overview section, you can filter the list of signal maps using context sensitive constraints. After selecting the search bar, you will be presented with a list of **Properties** to filter by. Selecting a property will present **Operators** such as Equals, Contains, Does not equal, and Does not contain. Selecting an operator will create a list of resources from the selected property type. Selecting one of these resources will cause the signal map list to only display signal maps that fit the constraint you defined.

Overview logs and metrics

To view CloudWatch metrics and logs for a signal map, select the radio button next to the name of the signal map. A tabbed interface for both metrics and logs will appear beneath the signal map list.

CloudWatch Metrics

CloudWatch metrics for the selected signal map will be context-sensitive and only display metrics associated with the services used in that signal maps workflow. You can use the on-screen metrics tools to customize the displayed metric periods and time ranges.

CloudWatch Logs

If you associated a CloudWatch log group with the signal map, that group will be displayed here.

Using workflow monitor signal maps

From the **overview** section of the console, you can select a specific signal map to view more information about that signal map and its attached monitoring resources.

After selecting a signal map, you will be presented with the signal map and a number of tabbed section containing more information:

- CloudWatch alarms
- EventBridge rules
- AWS Elemental alerts
- Metrics
- Logs
- Basic details

Navigating the signal map

A signal map will contain nodes for every supported AWS resource discovered by workflow monitor. Certain resources, such as MediaLive channels and MediaPackage endpoints can display thumbnail previews of the content, if thumbnail previews are available.

Selecting a resource node, and selecting **View selected resource details** from the **Actions** dropdown menu will take you to the associated service's details page. For example, selecting a MediaLive channel and selecting **View selected resource details** will open the MediaLive console's details page for that channel.

Selecting a resource node will filter the list of active alarms to only that node. If you select the resource's **Target ARN** in the active alarm, you will be taken to the associated service's details page, with the selected resource open.

Monitoring AWS Elemental Link hardware devices in AWS Elemental MediaLive

You can monitor AWS Elemental Link activity on the AWS Elemental MediaLive console.

Topics

Monitoring AWS Elemental Link with thumbnails

You can view display thumbnails of the content that is currently being pushed to MediaLive by an AWS Elemental Link hardware device. The thumbnails appear if the AWS Elemental Link hardware is pushing content. You don't have to have an input or a channel that is using this content.

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Input devices**, find the card for the Link input device that you want. If there are many Link input devices, enter part of the name to filter the list.

   The card shows a thumbnail panel. If the device is pushing content and the device is connected to AWS (as shown in the **Connection state** field), the thumbnail refreshes every 5 seconds.

Input device metrics for AWS Elemental MediaLive

You can use Amazon CloudWatch metrics to monitor input devices such as AWS Elemental Link. CloudWatch collects raw data that from these input devices, and processes it into readable, near real-time metrics that are kept for 15 months. You use CloudWatch to view the metrics. Metrics can help you gain a better perspective about how MediaLive is performing over the short term and long term.

Dimensions for input devices

- InputDeviceId – This value is a unique identifier for each input device.
- Device Type – The specific model type of the input device, such as AWS Elemental Link HD or UHD.

Topics

Using SDI

Indicates if SDI is the currently selected input for an AWS Elemental Link device.

A value of 0 indicates that SDI is not the active input. A value of 1 indicates that SDI is the active input.

Details:

- Name: UsingSdi
- Units: Boolean.
- Meaning of zero: SDI is not the selected input.
- Meaning of no datapoints: The device is not connected to AWS.
- Supported dimension sets: InputDeviceId and DeviceType.
- Recommended statistic: Minimum (SDI input is inactive) or maximum (SDI input is active).

Using HDMI

Indicates if HDMI is the currently selected input for an AWS Elemental Link device.

A value of 0 indicates that HDMI is not the active input. A value of 1 indicates that HDMI is the active input.

Details:

- Name: UsingHdmi
- Units: Boolean.
- Meaning of zero: HDMI is not the selected input.
- Meaning of no datapoints: The device is not connected to AWS.
- Supported dimension sets: InputDeviceId and DeviceType.
- Recommended statistic: Minimum (HDMI input is inactive) or maximum (HDMI input is active).

Input locked

Indicates if an AWS Elemental Link device has successfully locked on to the input signal.

A value of 0 indicates that the input signal is not locked. A value of 1 indicates that the input is successfully locked.

Details:

- Name: InputLocked
- Units: Boolean.
- Meaning of zero: The device is not locked onto a signal. Either because nothing is plugged in, or the device is unable to detect the input signal.
- Meaning of no datapoints: The device is not connected to AWS.
- Supported dimension sets: InputDeviceId and DeviceType.
- Recommended statistic: Maximum. Indicates that the signal is successfully locked.

Encoder running

The encoder is successfully processing the input signal from an AWS Elemental Link device

A value of 0 indicates that the encoder is not running and the input is not being processed. A value of 1 indicates that the encoder is successfully processing the locked input signal.

Details:

- Name: EncoderRunning
- Units: Boolean.
- Meaning of zero: The encoder is not processing the input signal. Verify that a valid signal is being passed to the input (device). Look at Locked and Running.
- Meaning of no datapoints: The device is not connected to AWS.
- Supported dimension sets: InputDeviceId and DeviceType.
- Recommended statistic: Maximum. Indicates that the encoder is successfully processing.

Linked to stream endpoint

An AWS Elemental Link device is connected to the streaming endpoint in AWS.

A value of 0 indicates that the device is not connected to the streaming endpoint. A value of 1 indicates that the device is successfully connected to the streaming endpoint.

Details:

- Name: LinkedToStreamEndpoint
- Units: Boolean.
- Meaning of zero: The device is not connected to a streaming endpoint.
- Meaning of no datapoints: The device is not connected to AWS.
- Supported dimension sets: InputDeviceId and DeviceType.
- Recommended statistic: Maximum. Indicates that the device is successfully connected to the streaming endpoint.

Streaming

An AWS Elemental Link device is successfully streaming the input signal to MediaLive.

A value of 0 indicates that the input signal is not being streamed through to MediaLive. A value of 1 indicates that the device is successfully streaming the input signal to MediaLive.

Details:

- Name: Streaming
- Units: Boolean.
- Meaning of zero: The device is not fully streaming. Verify that the previous metrics are displaying the recommended statistics.
- Meaning of no datapoints: The device is not connected to AWS.
- Supported dimension sets: InputDeviceId and DeviceType.
- Recommended statistic: Maximum. Indicates that the device is successfully streaming the input signal to MediaLive.

Temperature

The temperature in degrees Celsius of an AWS Elemental Link device. Consult your device's documentation for recommended operating conditions.

Details:

- Name: Temperature
- Units: Degrees Celsius.
- Meaning of zero: A temperature of zero degrees Celsius is below the recommended operating temperature of the AWS Elemental Link device family.
- Meaning of no datapoints: The device is not connected to AWS.
- Supported dimension sets: InputDeviceId and DeviceType.
- Recommended statistic: Average.

Configured bitrate

The configured **Maximum bitrate** on an AWS Elemental Link device.

This value represents the highest bitrate that the input signal will be encoded.

Details:

- Name: ConfiguredBitrate
- Units: Bits per second.

- Meaning of zero: Not applicable. Consult the minimum required bitrates for your device.
- Meaning of no datapoints: The device is not connected to AWS.
- Supported dimension sets: InputDeviceId and DeviceType.
- Recommended statistic: P90.

Encoder bitrate

The actively encoded bitrate on an AWS Elemental Link device

This value represents the actual bitrate that is being encoded. If a **Maximum bitrate** has been configured (this is represented by the **Configured bitrate** value), this value will not exceed it.

Details:

- Name: EncoderBitrate
- Units: Bits per second.
- Meaning of zero: The encoder is not running.
- Meaning of no datapoints: The device is not connected to AWS.
- Supported dimension sets: InputDeviceId and DeviceType.
- Recommended statistic: P90.

Configured bitrate available

On an AWS Elemental Link device, the portion of **Configured bitrate** that the device can satisfy based on network conditions.

The bitrate that is actively encoded is a result of the configured bitrate and the network conditions at the time the metric is measured.

If a **Maximum bitrate** is configured, the input device will evaluate the network connection and deliver at a bitrate below the maximum, as long as the network supports it. If no **Maximum bitrate** value is set, the input device will determine the best bitrate for the network connection between the device and the MediaLive service. The **Encoder bitrate** metric represents the actual encoding bitrate, whether a **Maximum bitrate** value is set, or isn't set.

Details:

- Units: Percentage.
- Meaning of zero: Not applicable. While the encoder is running, a non-zero bitrate will be encoded.
- Meaning of no datapoints: The device is not streaming.
- Supported dimension sets: InputDeviceId and DeviceType.
- Recommended statistic: P90.

Total packets

On an AWS Elemental Link device, the total number of packets that were successfully delivered to the AWS streaming endpoint.

Details:

- Units: Count.
- Meaning of zero: No packets are being delivered from the Link device to the streaming endpoint.
- Meaning of no datapoints: The device is not streaming.
- Supported dimension sets: InputDeviceId and DeviceType.
- Recommended statistic: Sum.

Recovered packets

On an AWS Elemental Link device, the number of packets that were lost during transit, but recovered by error correction.

Details:

- Name: RecoveredPackets
- Units: Count.
- Meaning of zero: The stream is healthy. Successfully delivered packets did not require error correction.
- Meaning of no datapoints: The device is not streaming.
- Supported dimension sets: InputDeviceId and DeviceType.
- Recommended statistic: Sum.

Not recovered packets

On an AWS Elemental Link device, the number of packets that were lost during transit and were not recovered by error correction.

Details:

- Name: NotRecoveredPackets
- Units: Count.
- Meaning of zero: The stream is healthy. No packets have been lost in transit from the Link device to the streaming endpoint.
- Meaning of no datapoints: The device is not streaming.
- Supported dimension sets: InputDeviceId and DeviceType.
- Recommended statistic: Sum.

Error seconds

On an AWS Elemental Link device, the number of seconds in which one or more packets were dropped and not recovered.

Details:

- Name: ErrorSeconds
- Units: Count.
- Meaning of zero: The stream is healthy. No packets have been lost in transit from the Link device to the streaming endpoint.
- Meaning of no datapoints: The device is not streaming.
- Supported dimension sets: InputDeviceId and DeviceType.
- Recommended statistic: Sum.

Use cases

Scenario: My device is not streaming.

If you have started a channel, but find that the stream is not functioning correctly, you can use metrics to isolate the source of the issue. The following metrics represent different points from

the input source to the final stream. An issue at any point could indicate why your stream is not working.

To find a malfunctioning stream, look at the following metrics (in order). Start at input source and end at final stream.

- **Using SDI/Using HDMI**
  - Verify the Link device is configured to use the input type that matches the connected source.

- **Input locked**
  - If this is 0, the encoder cannot identify a signal from the connected source. Verify that you have a connected source that matches the selected input type.

- **Encoder running**
  - If this is 0, the Link device cannot encode the signal. If the input is locked, this could indicate a problem with the Link device.

- **Linked to stream endpoint**
  - If this is 0, the Link device cannot connect to its streaming endpoint in the AWS service. Verify the encoder is running by checking the **Encoder running** metric. If the encoder is running, verify that port 2088 is not blocked on your network. For a list of ports that must be open, consult the HD datasheet or UHD datasheet.

- **Streaming**
  - If this is 0, verify that the channel has been started. If the value is still 0, investigate the previous metrics to isolate the source of the problem.

Scenario: My video quality is substandard.

Substandard video quality can be the result of a network performance issue. To determine if network performance is the cause, look at **Configured bitrate**, **Encoder bitrate**, and **Configured bitrate available**. If **Configured bitrate available** is consistently less than 100%, that indicates that the Link device's network connection is unable to satisfy the configured bandwidth. When that happens, the **Encoder bitrate** is reduced to adjust to the inferior network connection.

When the **Encoder bitrate** is reduced due to a network connection issue, the encoder attempts to maintain video quality by preventing packet loss. However, the resolution, frame rate, and scene complexity can affect the encoder's ability to produce a high quality stream. We recommend that HD devices running at 60 frames per second (FPS) maintain an **Encoder bitrate** of at least five Megabits per second (Mbps). UHD devices running at 60 FPS should maintain an **Encoder bitrate** of 10 to 15 Mbps.

You can use the following metrics to troubleshoot the frequency and severity of network interruptions:

- **Recovered packets**
  - If this is greater than 0, packets were dropped in transit and were recovered by error correction. Although recovered packets will not impact video quality, consistent packet drops can indicate that the stream might experience issues in the future.
- **Not recovered packets**
  - If this is greater than 0, packets were dropped in transit and were not recovered by error correction. A loss of packets can result in poor video quality. You can compare this with the value of **Total packets** to determine what percentage of incoming packets were lost.
- **Error seconds**
  - If this is greater than 0, it indicates that the stream experienced one or more seconds in which packets were dropped and not recovered. This metric quantifies video quality issues as a total period of impacted time, rather than a packet count.

Managing channel maintenance

The AWS Elemental MediaLive service routinely performs maintenance on underlying systems for security, reliability, and operational performance. The maintenance activities include actions such as patching the operating system, updating drivers, or installing software and patches.

Maintenance is performed individually on each channel, as it is required.

You can't disable channel maintenance. But you can control when the maintenance occurs.

The routine for maintenance is the following:

- When you create a channel, MediaLive automatically assigns an arbitrary maintenance window: a particular day of the week and a two-hour window. For example, Thursdays from 4:00 to 5:00 UTC.
- When a channel needs maintenance, you receive notification in the AWS Health Dashboard and by email. For more information, see the section called “Managing notifications”.
- When you receive a notification, you should decide if you want to adjust the timing of the maintenance. There are several ways to adjust the timing. See the section called “Options for handling maintenance”.

Topics

Viewing maintenance information

Viewing maintenance information in MediaLive

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Channels**.

In the list of channels that appears, there are two columns on the right-hand side: **Maintenance status** and **Maintenance window**, which shows the upcoming maintenance event (if any).

Viewing maintenance information on the Personal Health Dashboard

On the Personal Health Dashboard, you can view information upcoming maintenance events for all channels in your AWS account.

1. Open the AWS Health Dashboard at https://phd.aws.amazon.com/phd/home#/.
2. In the navigation pane, choose **Your account health**, then choose **Other notifications**. Use the filter to find events with a title that includes **MediaLive maintenance event**.

   Each event lists the channels, the Region, and the state date.

Managing maintenance notifications

When a channel needs maintenance, you receive notification in the AWS Health Dashboard, one notification for each channel. In addition, AWS sends an email to the email address associated with your AWS account.

We recommend that you can set up EventBridge to disseminate these notifications to people in your organization. See the *AWS Health Dashboard user guide* for more information. The people who receive those notifications should read this maintenance topic.

Working with a maintenance event

You receive notification of upcoming maintenance for a channel at least 21 calendar days before the deadline for the maintenance. The notification specifies this deadline. You should decide how you want to handle the upcoming maintenance event.

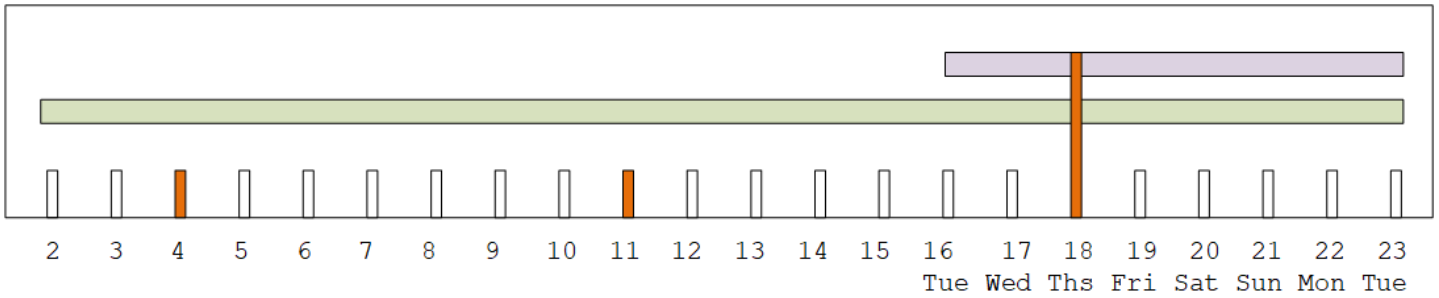
Topics

How maintenance timing works

In the following example, assume that your *maintenance window* is currently set for Thursdays between 4:00 and 5:00 UTC (red date marks in the diagram). Assume that you receive a *maintenance notification* on Tuesday, May 2.

- The *maintenance deadline* is Tuesday, May 23.
- The green bar is the current *maintenance event period*. It is the period between the notification and the deadline. In this example, the maintenance event period is May 2 to May 23.
- The purple bar is the *maintenance opening*. It is the period from 7 days before the deadline until the deadline. In this example, the maintenance opening is May 16 to May 23.

- The short red marks are *potential maintenance events*. Each potential maintenance event is set on the same day. In this example, there is a potential maintenance event every Thursday.
- The red mark in the purple bar is the *current maintenance window*. Automatic maintenance is set to occur some time during the maintenance window that occurs during the maintenance opening. In this example, it is set to occur on Thursday, May 18 between 4:00 and 5:00 UTC.



Options for handling maintenance

You have the following options for maintenance:

- You can leave the maintenance window (red mark) as it is currently set.
- You can change the day of the week and the time of the maintenance window. See the section called “Change the maintenance window”.
- You can set a specific date and time for the maintenance window. See the section called “Set a specific date”

Stopping a channel during the maintenance event period

As part of normal operations, you might stop the channel, for example, to make changes to the channel configuration.

If you stop a channel during the maintenance event period (green bar), maintenance will be performed automatically when you restart. The maintenance event will be considered to be completed. The maintenance status for the channel will change to **Not required**.

Rescheduling a maintenance event

If MediaLive can't perform the maintenance during the maintenance window (red mark), MediaLive will reschedule the maintenance for the same maintenance window in the next week. This date might be after the deadline for the maintenance event period (green bar). Every week, MediaLive will try to perform the maintenance.

Each time MediaLive reschedules the maintenance event, the new date will appear in the Channels list in the MediaLive console, and on the AWS Health Dashboard.

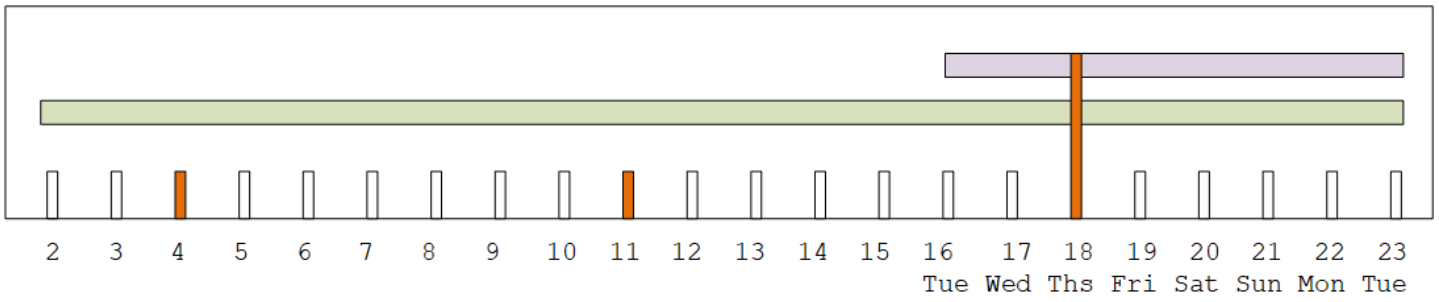
During this retry period, you can change the maintenance window, but only if the channel is still in the maintenance event period (green bar).

Changing the maintenance window

There are two ways to change the maintenance window (red mark): editing the maintenance window, and setting a specific date. The method to choose depends on your reasons for wanting the change. The following table compares the reason and period for the two methods. Read across each row to compare the two methods.

	Edit the maintenance window	Set a specific date
Reason for change	Use this method if you're happy to wait until the next maintenance opening (purple bar), but the current day of the week and/or time doesn't suit your operations.	Use this method if you don't want to wait until the next maintenance opening (purple bar) for maintenance. You want to move the maintenan ce window earlier in the

	Edit the maintenance window	Set a specific date
		maintenance event period (green bar).
Period when you can make the change	Any time from the minute that you create the channel until one minute before the start of the upcoming maintenance window (red mark).	From the start of the maintenance event period (green bar) until one minute before the start of the upcoming maintenance window (red mark). You can't change the maintenance window outside of the maintenance event period.



Change the maintenance window

You can change the current maintenance window (red mark). The following rules apply:

- You can change the window at any time from the minute that you create the channel until one minute before the maintenance window in the current or next maintenance event. Therefore, following our example, you can change the window at any time until Thursday, May 18 at 3:59 UTC.
- The new window applies to future maintenance events, not just to the next maintenance event. The maintenance window will change, for example, from every Thursday to every Saturday.

- You can move the maintenance window to an earlier day or a later day of the week. Maintenance will occur in that window during the maintenance opening (purple bar). For example, you can change the maintenance window to Saturdays, at 03:00 UTC. This specific maintenance event will occur on Saturday, May 20, some time from 03:00 to 05:00 UTC.
- You can't change the window if doing so would mean that the current maintenance event won't occur. For example, on Thursday, May 17 at 1:00 UTC you can't change the window to Wednesdays because the next Wednesday is May 24, which is after the deadline.

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Channels**, and select one or more channels. Select only channels that have the status **Maintenance Required**.
3. Choose **Channel actions**, and then choose **Edit channel maintenance window**.
4. On the dialog that appears, set **Start day** and **Start hour**. Choose **Save**.

Set a specific date

You can set up a specific date and time for the maintenance window (red mark). The following rules apply:

- You can change the window at any time from the start of the maintenance event period (green bar) until one minute before the start of the current maintenance window. Therefore, following our example, you can change the window at any time from 0.01 UTC on May 2 until Thursday, May 18 at 3:59 UTC.
- The specific date and time can be any time in the maintenance event period (green bar), so long as the new date is still in the future.
- This action sets a specific date for the maintenance, and it also changes the maintenance window changes to the day of the week of the specific date, and the time of the specific date. For example, if you specify Tuesday, May 9 at 2:00 UTC, then the maintenance window changes permanently to Tuesdays at 2:00 UTC.

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Channels**, and select one or more channels. Select only channels that have the status **Maintenance Required**.
3. Choose **Channel actions**, and then choose **Edit channel maintenance window**.

4. On the dialog that appears, set a **Start hour**. Ignore **Start date**.
5. Expand **Additional maintenance settings** in the **Upcoming maintenance** section. In **Maintenance window date**, set the specific date. Choose **Save**.

How MediaLive performs channel maintenance

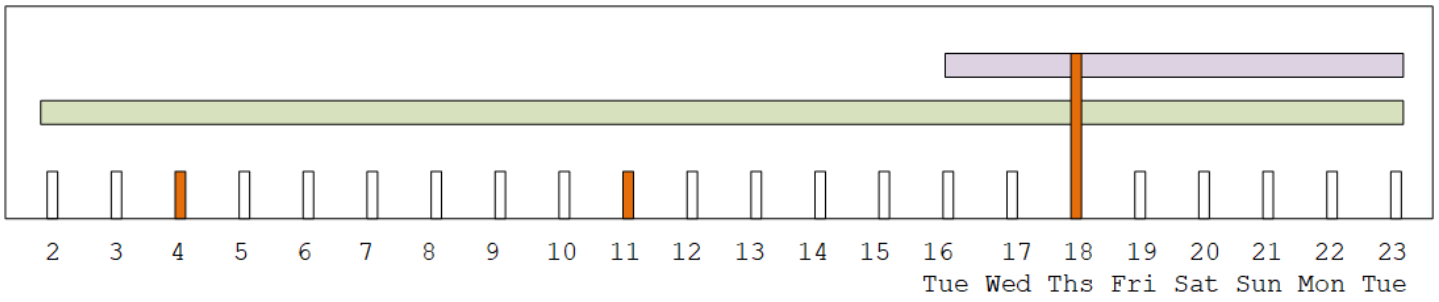
At some point during the maintenance window (red mark), MediaLive starts the maintenance. There is no notification that maintenance is about to start on the channel.

There is no need to monitor the channel or to prepare for maintenance during the time leading up to the maintenance window.

MediaLive performs maintenance as follows:

- If a channel is set up as a standard channel (with two pipelines), MediaLive always performs maintenance on one pipeline at a time. MediaLive stops one pipeline, performs the maintenance, and automatically restarts the pipeline. It then stops the second pipeline, performs the maintenance, and automatically restarts the second pipeline. In this way, there is typically no impact on the output from the channel.
- If a channel is set up as a single-line channel, MediaLive stops the pipeline, which stops the channel. MediaLive performs the maintenance and restarts the channel. There will be no output from the channel while maintenance is being performed.

> **Note**
>
> Setting up with standard channels is an effective way to mitigate the impact of maintenance events. You might want to consider this mitigation for your most important 24x7 channels.



AWS Elemental MediaLive Reference

This chapter contains reference tables for different features and functions of AWS Elemental MediaLive.

Topics

Supported captions

This section contains tables that specify the caption formats that are supported in inputs and the caption formats that are supported in outputs.

There are several factors that control your ability to output captions in a given format:

- The type of input container. A given input container can contain captions in some formats and not in others.
- The format of the input captions. A given format of captions can be converted to some formats and not to others.
- The type of output containers. A given output container supports some caption formats and not others.

Topics

- Formats supported in an HLS output or a MediaPackage output
- Formats supported in a Microsoft Smooth output
- Formats supported in an RTMP output
- Formats supported in a UDP output or a multiplex output

General information about supported formats

The following table shows the supported formats, specifies whether they are supported in inputs or outputs, and specifies the standard that defines each format.

Caption	Supported in input	Supported in output	Description
Ancillary data	Yes		Data that is compliant with "*SMPTE 291M: Ancillary Data Packet and Space Formatting*" and that is contained in ancillary data.
ARIB	Yes	Yes	Captions that are compliant with *ARIB STD-B37 Version 2.4*.
Burn-in		Yes	From input: It is technically impossible for the encoder to read burn-in captions. Therefore, from an input viewpoint, they can't be considered to be captions. For output: Burn-in captions are captions

Caption	Supported in input	Supported in output	Description
			that are converted into text and then overlaid on top of the picture directly in the video stream.
DVB-Sub	Yes	Yes	Captions that are compliant with ETSI EN 300 743.
EBU-TT-D		Yes	Captions that are compliant with EBU Tech 3380, EBU-TT-D Subtitling Distribution Format, 2018.

Caption	Supported in input	Supported in output	Description
Embedded	Yes	Yes	In most containers: Captions that are compliant with the EIA-608 standard (also known as CEA-608 or *line 21 captions*) or the CEA-708 standard (also known as EIA-708). In a Link input container: Captions carried as ancillary captions that are compliant with SMPTE 334. The ancillary captions are compliant with EIA-608 standard (also known as CEA-608 or *line 21 captions*) or CEA-708 standard (also known as EIA-708).
Embedded+SCTE-20	Yes	Yes	Captions that have both embedded and SCTE-20 in the video. The embedded captions are inserted before the SCTE-20 captions.

Caption	Supported in input	Supported in output	Description
RTMP CaptionInfo		Yes	Captions that are compliant with the Adobe onCaptionInfo format.
SCTE-20	Yes		Captions that are compliant with the standard "*SCTE 20 2012 Methods for Carriage of CEA-608 Closed Captions and Non-Real Time Sampled Video*."
SCTE-20+Embedded		Yes	Captions that are compliant with SCTE-43. The SCTE-20 captions are inserted in the video before the embedded captions.
SCTE-27	Yes		Captions that are compliant with the standard "*SCTE-27 (2011), Subtitlin g Methods for Broadcast Cable*."
SMPTE-TT		Yes	Captions that are compliant with the standard "*SMPTE ST 2052-1:2010*".

Caption	Supported in input	Supported in output	Description
Teletext	Yes	Yes	From TS input: Captions in the EBU Teletext format. From a CDI input or in a Link container : Captions in OP47 teletext format, also known as SMPTE RDD-08 (compliant with ITU-R BT.1120-7).
TTML		Yes	Captions files that are compliant with the standard "*Timed Text Markup Language 1 (TTML1) (Second Edition)*."
WebVTT		Yes	Captions that are compliant with "*webvtt: The Web Video Text Tracks Format*" (http://dev.w3.org/html5/webvtt/).

Captions categories

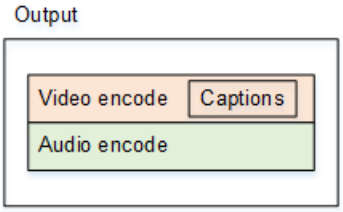
Captions are grouped into five categories, based on how the captions are included in the output.

Captions format	Category of this format
ARIB	Object-style

Captions format	Category of this format
Burn-in	Burn-in
DVB-Sub	Object-style
EBU-TT-D	Sidecar
Embedded	Embedded
Embedded+SCTE-20	Embedded
RTMP CaptionInfo	Object-style
SCTE-20+Embedded	Embedded
SCTE-27	Object-style
SMPTE-TT	Stream
Teletext	Object-style
TTML	Sidecar
WebVTT	Sidecar

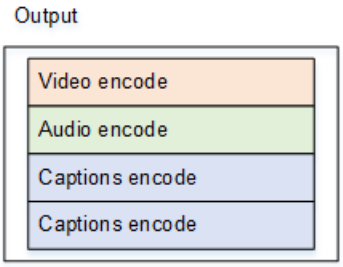
Embedded captions

The captions are carried inside the video encode, which is itself in an output in the output group. There is only ever one captions entity within that video encode, although that entity might contain captions for up to four languages.



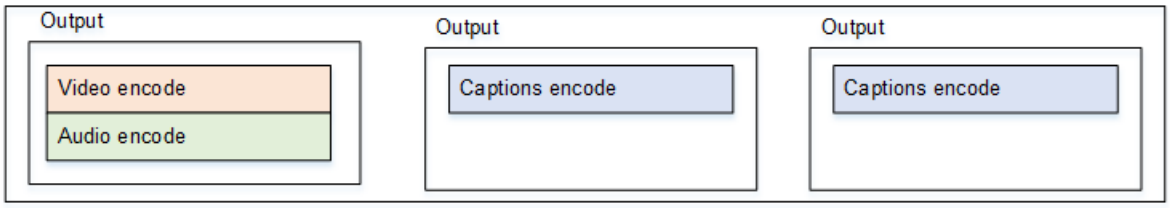
Object-style captions

All the captions encodes for a given output group are in the same output as the corresponding video and audio.



Sidecar captions

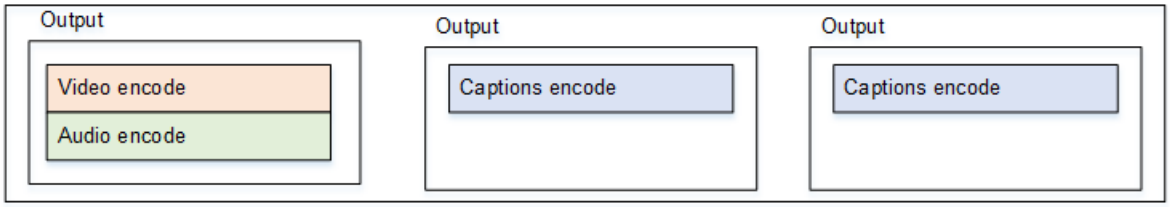
Each captions encode for a given output group is in its own "captions-only" output. The output group can contain more than one captions output, for example, one for each language.



Each captions-only output becomes a separate file in the packaged output.

Stream

Each captions encode for a given output group is in its own "captions-only" output. The output group can contain more than one captions output, for example, one for each language.



Each captions-only output becomes a separate stream in the packaged output.

Burn-in captions

The captions are converted into text and then overlaid on the picture directly in the video encode. Strictly speaking, once the overlay occurs, these are not really captions because they are indistinguishable from the video.

How to read the supported captions information

With captions, there are constraints on the ability to produce a specific output format from the input format.

You must make sure it is possible to produce the output formats that you want in a specific output type, from the captions in the input. For example, you must make sure that you can produce DVB-Sub captions in an Archive output, when the source is an HLS input that contains SCTE-20 captions.

To determine that the input type and input captions format can produce the chosen captions format in the chosen output type, consult the tables in the following sections.

Follow these steps

1. Find the table for your output container. For example, Archive.
2. In that table, look in the first column for the container type of the input that you have been provided with. For example, HLS.
3. In the second column, find the input captions that are in that container. For example, SCTE-20.
4. In the third column, look for the output captions format that you require. For example, DVB-Sub.

   If the format is listed, then your input is suitable.

   If the format is not listed, you must ask the provider of that input to provide a different source.

The tables for the supported formats are in the following sections:

- the section called “Archive output”
- the section called “HLS or MediaPackage output”
- the section called “Microsoft Smooth output”
- the section called “RTMP output”
- the section called “UDP or multiplex output”

Formats supported in an Archive output

In this table, look up your input container and captions type. Then read across to find the caption formats that are supported in an Archive (MPEG-TS file) output, when you have this input container and captions type.

Source caption container	Source caption input	Supported output captions
CDI container	ARIB	ARIB
	Embedded	Burn-in DVB-Sub Embedded Embedded+SCTE-20 SCTE-20
	Teletext	DVB-Sub Teletext
HLS container	Embedded	Burn-in DVB-Sub Embedded Embedded+SCTE-20 SCTE-20 SCTE-20+Embedded
	SCTE-20	Burn-in DVB-Sub Embedded Embedded+SCTE-20 SCTE-20 SCTE-20+Embedded
Link container	Embedded	Burn-in

Source caption container	Source caption input	Supported output captions
		DVB-Sub Embedded Embedded+SCTE-20 SCTE-20 SCTE-20+Embedded
	Teletext	DVB-Sub Teletext
MP4 container	Ancillary	Burn-in DVB-Sub Embedded Embedded+SCTE-20 SCTE-20 SCTE-20+Embedded
	Embedded or Embedded+ SCTE-20	Burn-in DVB-Sub Embedded Embedded+SCTE-20 SCTE-20 SCTE-20+Embedded

Source caption container	Source caption input	Supported output captions
RTMP container	Embedded	Burn-in DVB-Sub Embedded Embedded+SCTE-20 SCTE-20 SCTE-20+Embedded
MPEG-TS container (through the RTP or MediaConnect protocol)	ARIB	ARIB
	DVB-Sub	Burn-in DVB-Sub
	Embedded or Embedded+ SCTE-20	Burn-in DVB-Sub Embedded Embedded+SCTE-20 SCTE-20 SCTE-20+Embedded
	SCTE-20	Burn-in DVB-Sub Embedded Embedded+SCTE-20 SCTE-20 SCTE-20+Embedded

Source caption container	Source caption input	Supported output captions
	SCTE-27	None
	Teletext	DVB-Sub Teletext

Formats supported in a CMAF Ingest output

In this table, look up your input container and captions type. Then read across to find the caption formats that are supported for a CMAF Ingest output, when you have this input container and captions type.

Source caption container	Source caption input	Supported output captions
CDI container	Embedded	Burn-in TTML
	Teletext	TTML
HLS container	Embedded	Burn-in TTML
	SCTE-20	Burn-in TTML
Link container	Embedded	Burn-in TTML
	Teletext	Burn-in TTML
MP4 container	Ancillary	Burn-in

Source caption container	Source caption input	Supported output captions
		TTML
	Embedded or Embedded+ SCTE-20	Burn-in TTML
RTMP container	Embedded	Burn-in TTML
MPEG-TS container (through the RTP or MediaConnect protocol)	Embedded or Embedded+ SCTE-20	Burn-in TTML
	SCTE-20	Burn-in TTML
	Teletext	Burn-in TTML

Formats supported in an HLS output or a MediaPackage output

In this table, look up your input container and captions type. Then read across to find the caption formats that are supported for an HLS output or MediaPackage output, when you have this input container and captions type.

Source caption container	Source caption input	Supported output captions
CDI container	ARIB	None
	Embedded	Burn-in Embedded WebVTT

Source caption container	Source caption input	Supported output captions
	Teletext	Teletext
HLS container	Embedded	Burn-in Embedded WebVTT
	SCTE-20	Burn-in Embedded WebVTT
Link container	Embedded	Burn-in Embedded WebVTT
	Teletext	Burn-in WebVTT
MP4 container	Ancillary	Burn-in Embedded WebVTT
	Embedded or Embedded+ SCTE-20	Burn-in Embedded WebVTT
RTMP container	Embedded	Burn-in Embedded WebVTT

Source caption container	Source caption input	Supported output captions
MPEG-TS container (through the RTP or MediaConnect protocol)	ARIB	None
	DVB-Sub	Burn-in WebVTT
	Embedded or Embedded+ SCTE-20	Burn-in Embedded WebVTT
	SCTE-20	Burn-in Embedded WebVTT
	SCTE-27	Burn-in WebVTT
	Teletext	Burn-in WebVTT

Formats supported in a Microsoft Smooth output

In this table, look up your input container and captions type. Then read across to find the caption formats that are supported for a Microsoft Smooth output, when you have this input container and captions type.

Source caption container	Source caption input	Supported output captions
CDI container	ARIB	None
	Embedded	Burn-in EBU-TT

Source caption container	Source caption input	Supported output captions
		SMPTE-TT
		TTML
	Teletext	Burn-in
		EBU-TT
		SMPTE-TT
		TTML
HLS container	Embedded	Burn-in
		EBU-TT-D
		SMPTE-TT
		TTML
	SCTE-20	Burn-in
		EBU-TT-D
		SMPTE-TT
		TTML
Link container	Embedded	Burn-in
		EBU-TT-D
		SMPTE-TT
		TTML
	Teletext	Burn-in
		SMPTE-TT
		TTML

Source caption container	Source caption input	Supported output captions
MP4 container	Ancillary	Burn-in EBU-TT-D SMPTE-TT TTML
	Embedded or Embedded+ SCTE-20	Burn-in EBU-TT-D SMPTE-TT TTML
RTMP container	Embedded	Burn-in EBU-TT-D SMPTE-TT TTML
MPEG-TS container (through the RTP or MediaConnect protocol)	ARIB	None
	DVB-Sub	SMPTE-TT
	Embedded or Embedded+ SCTE-20	Burn-in EBU-TT-D SMPTE-TT TTML

Source caption container	Source caption input	Supported output captions
	SCTE-20	Burn-in EBU-TT-D SMPTE-TT TTML
	SCTE-27	Burn-in SMPTE-TT
	Teletext	Burn-in EBU-TT-D SMPTE-TT TTML

Formats supported in an RTMP output

In this table, look up your input container and captions type. Then read across to find the caption formats that are supported for an RTMP output, when you have this input container and captions type.

Source caption container	Source caption input	Supported output captions
CDI container	ARIB	None
	Embedded	Burn-in Embedded RTMP CaptionInfo
	Teletext	None
HLS container	Embedded	Burn-in

Source caption container	Source caption input	Supported output captions
		Embedded RTMP CaptionInfo
	SCTE-20	Embedded
Link container	Embedded	Burn-in Embedded RTMP CaptionInfo
	Teletext	None
MP4 container	Ancillary	Burn-in Embedded RTMP CaptionInfo
	Embedded or Embedded+ SCTE-20	Burn-in Embedded RTMP CaptionInfo
RTMP container	Embedded	Burn-in Embedded RTMP CaptionInfo
MPEG-TS container (through the RTP or MediaConnect protocol)	ARIB	None
	DVB-Sub	Burn-in
	Embedded or Embedded+ SCTE-20	Burn-in Embedded RTMP CaptionInfo

Source caption container	Source caption input	Supported output captions
	SCTE-20	Embedded RTMP CaptionInfo
	SCTE-27	Burn-in
	Teletext	None

Formats supported in a UDP output or a multiplex output

In this table, look up your input container and captions type. Then read across to find the caption formats that are supported for an MPEG-TS streaming output over UDP or RTP, or for an MPTS multiplex output, when you have this input container and captions type.

Source caption container	Source caption input	Supported output captions
CDI container	ARIB	ARIB
	Embedded	Burn-in DVB-Sub Embedded Embedded+SCTE-20 SCTE-20
	Teletext	Burn-in DVB-Sub Teletext
HLS container	Embedded	Burn-in DVB-Sub Embedded

Source caption container	Source caption input	Supported output captions
		Embedded+SCTE-20 SCTE-20 SCTE-20+Embedded
	SCTE-20	Burn-in DVB-Sub Embedded Embedded+SCTE-20 SCTE-20 SCTE-20+Embedded
Link container	Embedded	Burn-in DVB-Sub Embedded Embedded+SCTE-20 SCTE-20 SCTE-20+Embedded
	Teletext	Burn-in DVB-Sub WebVTT

Source caption container	Source caption input	Supported output captions
MP4 container	Ancillary	Burn-in DVB-Sub Embedded Embedded+SCTE-20 SCTE-20 SCTE-20+Embedded
	Embedded or Embedded+ SCTE-20	Burn-in DVB-Sub Embedded Embedded+SCTE-20 SCTE-20 SCTE-20+Embedded
RTMP container	Embedded	Burn-in DVB-Sub Embedded Embedded+SCTE-20 SCTE-20 SCTE-20+Embedded
MPEG-TS container	ARIB	ARIB
(through the RTP or MediaConnect protocol)	DVB-Sub	Burn-in DVB-Sub

Source caption container	Source caption input	Supported output captions
	Embedded or Embedded+ SCTE-20	Burn-in DVB-Sub Embedded Embedded+SCTE-20 SCTE-20 SCTE-20+Embedded
	SCTE-20	Burn-in DVB-Sub Embedded Embedded+SCTE-20 SCTE-20 SCTE-20+Embedded
	SCTE-27	None
	Teletext	Burn-in DVB-Sub Teletext

Input types in AWS Elemental MediaLive

This section provides lookup information about the types of sources that AWS Elemental MediaLive can ingest.

Topics

Input types supported in AWS Elemental MediaLive

MediaLive supports the following input types. For more information about these types, see the sections that follow.

- CDI
- HLS
- Link
- MediaConnect
- MP4
- TS
- RTMP Pull
- RTMP Push
- RTP

Input types, protocols, and upstream systems in AWS Elemental MediaLive

The following table lists the supported input types, and describes how the input handles the source content. In the table, find a type of input in the first column, then read across the row for information about how the type is supported.

The sections after the table describe how MediaLive ingests a push or pull input.

MediaLive input type	Push or pull?	Use case	Upstream system and supported protocol
CDI	Push	Push an uncompressed stream in your VPC to a fixed endpoint on MediaLive.	Amazon VPC within a private cloud
HLS See HLS inputs, after this table.	Pull	Pull an HLS stream or asset from an external endpoint using the HTTP protocol, with or without a secure connection.	HTTP server or HTTPS server
HLS		Pull an HLS stream or VOD asset from an AWS Elemental MediaStore container, using a secure connection.	AWS Elemental MediaStore with a custom protocol
HLS		Pull an HLS stream or VOD asset from an Amazon S3 bucket, using a secure connection.	Amazon S3 over a custom protocol
Link	Push	Push a transport stream (TS) from an AWS Elemental Link device.	AWS Elemental Link over an internal connection
MediaConnect	Push	Push a transport stream (TS) from	AWS Elemental MediaConnect over

MediaLive input type	Push or pull?	Use case	Upstream system and supported protocol
		a flow in AWS Elemental MediaConnect. This input uses a MediaConnect flow ARN, not a URI.	an internal connection
MP4	Pull	Pull an MP4 file from an HTTP server, with or without a secure connection.	HTTP server or HTTPS server
MP4		Pull an MP4 file from an Amazon S3 bucket, using a secure connection. With MediaLive, the bucket name can't use *dot* notation. For example, `mycompany-videos` is valid, but `mycompany.videos` isn't.	Amazon S3 over a custom protocol

MediaLive input type	Push or pull?	Use case	Upstream system and supported protocol
Transport Stream (TS) file	Pull	Pull a TS file from an HTTP server, with or without a secure connectio n. File extensions of .ts or .m2ts are supported. MediaLive only supports static TS files. It does not support growing TS files that are written to while in use.	HTTP server or HTTPS server

MediaLive input type	Push or pull?	Use case	Upstream system and supported protocol
Transport Stream (TS) file		Pull a TS file from an Amazon S3 bucket, using a secure connection. File extensions of .ts or .m2ts are supported. MediaLive only supports static TS files. It does not support growing TS files that are written to while in use. With MediaLive, the bucket name can't use *dot* notation. For example, `mycompany-videos` is valid, but `mycompany.videos` isn't.	Amazon S3 over a custom protocol
RTMP Pull	Pull	Pull a stream from an external endpoint using the RTMP protocol. MediaLive doesn't support inputs using the RTMPS protocol.	RTMP server over RTMP Pull

MediaLive input type	Push or pull?	Use case	Upstream system and supported protocol
RTMP Push	Push	Push a stream to a fixed endpoint on MediaLive using the RTMP protocol. MediaLive doesn't support inputs using the RTMPS protocol.	RTMP server over RTMP Push
RTMP Push		Push a stream in your VPC to a fixed endpoint on MediaLive, using the RTMP protocol. MediaLive doesn't support inputs using the RTMPS protocol.	Amazon VPC over RTMP within a private cloud
RTP	Push	Push a transport stream (TS) to a fixed endpoint on MediaLive, using the RTP protocol.	RTP server over RTP Push
RTP		Push a transport stream (TS) in your VPC to a fixed endpoint on MediaLive, using the RTP protocol.	Amazon VPC over RTP within a private cloud

HLS

With HLS, the media must be a transport stream. MediaLive doesn't support ingest of HLS content that contains any other type of media.

Ingesting with a pull input

A pull input works as follows: the source continually publishes to an endpoint that is outside of MediaLive. When the channel (that is connected to the input) is running, MediaLive connects to the input and ingests the content.

When the channel is not running, MediaLive does not connect to the input. (There might be other applications that do connect.)

A pull input works with a streaming input (where the source is continually being published) or a VOD input (where the source is made available on the endpoint and then does not change).

Ingesting with an RTMP push input

An RTMP push input works as follows: the source attempts to deliver to an endpoint that is specified in the MediaLive input. There must be a handshake between the source and the MediaLive channel so that the source has information about the status of the input.

When the channel (that is connected to this input) is started, MediaLive responds to the handshake message and ingests it. When the channel is not running, MediaLive does not react; the source goes into a paused state.

A push input works only with a streaming source.

Ingesting with an RTP push input

An RTP push input works as follows: the source attempts to deliver to an endpoint that is specified in the MediaLive input. The source is unaware of whether the content is being ingested by the MediaLive channel.

When the channel (that is connected to this input) is started, MediaLive reacts to the source and ingests it. When the channel is not running, MediaLive does not react; the source continues to publish to the endpoint, but MediaLive ignores that action.

A push input works only with a streaming source.

Support for live and file sources in AWS Elemental MediaLive

The following table specifies whether an input type supports live streams or VOD assets.

MediaLive input type	Live stream supported?	VOD asset supported?
CDI	Yes	No
HLS from an HTTP or HTTPS server, or from MediaStore	Yes MediaLive considers an HLS input to be a *live stream* if the **Buffer segments** field has a value from 3 to 10, inclusive. (To display this field in the **Channel** page, in **General input settings** for **Network input settings**, choose **Network input**. For HLS input settings, choose **Hls input**. The **Buffer segments** field appears.)	Yes MediaLive considers the input to be a *VOD asset* if the **Buffer segments** field has a value of 11 or more, or is undefined (empty).
HLS from Amazon S3	Yes, as defined in the previous row We don't recommend Amazon S3 as a source for a live stream.	Yes, as defined in the previous row
Link	Yes	No
MediaConnect	Yes	No
MP4	No	Yes, with .mp4 file extension only
Transport Stream (TS) file	No	Yes, with .ts and .m2ts file extensions only
RTMP Pull	Yes	Yes

MediaLive input type	Live stream supported?	VOD asset supported?
RTMP Push	Yes	No
RTP	Yes	No

Supported input class in AWS Elemental MediaLive

There are two kinds of class for inputs — standard class inputs and single-class inputs. Some inputs can be set up as either input class. Some inputs can be set up only as the single-class. The class type to use depends on whether you plan to implement pipeline resiliency. For more information, see the section called “Channel classes and input classes”.

MediaLive input type	Can be set up as a single-class input	Can be set up as a standard-class input
CDI	No	Yes
HLS	Yes	Yes
Link	Yes	Yes
MediaConnect	Yes	Yes
MP4	Yes	Yes
Transport Stream (TS) file	Yes	Yes
RTMP Pull	Yes	Yes
RTMP Push	Yes	Yes
RTP	No	Yes

Support for setup as a VPC input in AWS Elemental MediaLive

Some inputs can be set up in Amazon Virtual Private Cloud (Amazon VPC). For more information, see the section called “Creating an input”.

MediaLive input type	Can be set up as a VPC input
CDI	Yes
HLS	No
Link	No
MediaConnect	No
MP4	No
Transport Stream (TS) file	No
RTMP Pull	No
RTMP Push	Yes
RTP	Yes

Input codecs in AWS Elemental MediaLive

This section provides information about the video and audio codecs that each input type can ingest in AWS Elemental MediaLive

Topics

Codecs supported in AWS Elemental MediaLive

MediaLive supports the following video codecs in sources:

- H.264 (AVC)
- H.265 (HEVC)
- MPEG-2

MediaLive supports the following audio codecs in sources:

- AAC
- Dolby Digital
- Dolby E wrapped in PCM
- Dolby Digital Plus
- MPEG Audio
- PCM

Supported codecs by input type in AWS Elemental MediaLive

The following table lists the video and audio codecs that each type of MediaLive input type supports.

Media type	Video codecs	Audio codecs
CDI See the section called "Characteristics for sources" for more information.	Uncompressed video	Dolby E wrapped in PCM PCM
HLS See HLS inputs, after this table.	H.264 (AVC)	AAC Dolby Digital Dolby Digital Plus
Link HD	Any codec that is included in a Link container is always supported by MediaLive.	Up to 8 channels of PCM audio when using HDMI or SDI input
Link UHD	Any codec that is included in a Link container is always supported by MediaLive.	Up to 8 channels of PCM audio when using HDMI input Up to 16 channels of PCM audio when using SDI input

Media type	Video codecs	Audio codecs
		Dolby Digital Dolby Digital Plus
MediaConnect	H.264 (AVC) H.265 (HEVC) MPEG-2	AAC Dolby Digital Dolby E wrapped in PCM Dolby Digital Plus MPEG Audio PCM
MP4	H.264 (AVC) H.265 (HEVC) MPEG-2	AAC Dolby E wrapped in PCM
Transport Stream (TS) file	H.264 (AVC) H.265 (HEVC) MPEG-2	AAC Dolby Digital Dolby E wrapped in PCM Dolby Digital Plus MPEG Audio PCM
RTMP	H.264 (AVC)	AAC

Media type	Video codecs	Audio codecs
RTP	H.264 (AVC) H.265 (HEVC) MPEG-2	AAC Dolby Digital Dolby E wrapped in PCM Dolby Digital Plus MPEG Audio PCM

HLS inputs

The audio and video assets can be multiplexed in a single stream. Or the audio can be in a separate audio rendition group. If you are using audio in a rendition group, the group can be selected by using the **Group ID** and **Name** that is in the **#EXT-X-MEDIA** tag.

Characteristics for video and audio sources in AWS Elemental MediaLive

Orientation

MediaLive only ingests landscape video. If a video source is configured as portrait, MediaLive will ingest it but will rotate it to landscape.

Input frame rate

MediaLive only supports constant frame rate (CFR) inputs. It does not support variable frame rates (VFR).

Other characteristics

Container	Video characteristics	Audio characteristics
CDI—MediaLive only supports these characteristics for CDI inputs.	• Uncompressed YCbCr 4:2:2 8-bit	• 24-bit Big-Endian PCM

Container	Video characteristics	Audio characteristics
	• Uncompressed YCbCr 4:2:2 10-bit	• Mono (1.0), Dual mono (2.0), Stereo (2.0), 5.1, 7.1 • 222, SGRP • 48kHz, 96 kHz

Supported output types in AWS Elemental MediaLive

This section provides information about the output content types and output video and codecs that MediaLive can produce.

Topics

List of output types supported in AWS Elemental MediaLive

MediaLive supports the following containers. Each of these containers is a different type of output group.

- Archive
- CMAF Ingest

- Frame Capture
- HLS
- MediaPackage
- Microsoft Smooth
- Multiplex
- RTMP
- RTMPS
- UDP

Containers, protocols, and downstream systems in AWS Elemental MediaLive

The following table lists the output formats and protocols that MediaLive supports. In the table, find a type of output in the first column, then read across the row for information about how the type is supported.

MediaLive output type (output group)	Use case	Downstream system and supported protocol	Live output supported	VOD output supported
Archive	Send transport stream (TS) files to an Amazon S3 bucket. See Amazon S3 Bucket Names, after this table.	Amazon S3, over a custom protocol	No	Yes. A channel can contain only one archive output group.
CMAF Ingest	Send a CMAF stream to a packager that supports that supports HTTPS PUT.	An HTTPS packager	Yes	No
Frame Capture	Send a series of JPEG files to an Amazon S3 bucket. See Amazon S3 Bucket Names, after this table.	Amazon S3, over a custom protocol	No	Yes. A channel can contain a maximum of three frame capture output groups.
HLS with a standard	Send an HLS stream to a	HTTP server	Yes	Yes, when the output group is

MediaLive output type (output group)	Use case	Downstream system and supported protocol	Live output supported	VOD output supported
container or an fMP4 container	server that supports HTTP PUT or WebDav.			set up for VOD mode
HLS with a standard container or an fMP4 container	Send an HLS stream to a server that supports HTTPS PUT or WebDav.	HTTPS server	Yes	Yes, when the output group is set up for VOD mode
HLS with a standard container or an fMP4 container	Send an HLS stream to an Akamai CDN.	Akamai CDN, over HTTP or HTTPS	Yes	No
HLS with a standard container only	Send an HLS stream to a MediaPackage channel using the HTTPS protocol.	AWS Elemental MediaPackage, over HTTPS with WebDAV	Yes	No
HLS with a standard container only	Send an HLS stream to a MediaPackage v2 channel using the HTTPS protocol.	AWS Elemental MediaPackage over HTTPS with Basic PUT	Yes	No
HLS with a standard container or an fMP4 container	Send an HLS stream to a container on MediaStore.	AWS Elemental MediaStore, with a custom protocol	Yes	Yes, when the output group is set up for VOD mode

MediaLive output type (output group)	Use case	Downstream system and supported protocol	Live output supported	VOD output supported
HLS with a standard container or an fMP4 container	Send an HLS stream to an Amazon S3 bucket. See Amazon S3 Bucket Names, after this table.	Amazon S3, over a custom protocol	Yes	Yes, when the output group is set up for VOD mode
MediaPackage	Send an HLS stream to a MediaPackage channel.	AWS Elemental MediaPackage over an HTTPS WebDav	Yes	No
Microsoft Smooth	Send a stream to an origin server or CDN that supports Microsoft Smooth Streaming.	A supported CDN, over HTTP or HTTPS	Yes	No
Multiplex	Create a transport stream (TS) that is part of a MediaLive multiplex.		Yes	No
RTMP	Send a stream to a server that supports the RTMP protocol.	RTMP server	Yes	No

MediaLive output type (output group)	Use case	Downstream system and supported protocol	Live output supported	VOD output supported
RTMPS	Send a stream to a server that supports the RTMPS protocol.	RTMPS server	Yes	No
UDP	Send a transport stream (TS) to a server that supports UDP.	UDP server	Yes	No

Amazon S3 bucket names

With MediaLive, the bucket name can't use *dot* notation. For example, `mycompany-videos` is valid, but `mycompany.videos` isn't.

Support for delivery in VPC in AWS Elemental MediaLive

The following table specifies which containers can be delivered to a destination in the VPC, when the channel that is set up for VPC delivery. For more information about VPC delivery, see the section called “VPC delivery”.

MediaLive output type (output group)	Can be delivered to a destination in your VPC	Can be delivered to a destination outside your VPC
Archive	A bucket, if Amazon S3 is set up with a VPC endpoint	Yes, if you associate Elastic IP addresses with the channel
CMAF Ingest	No	Yes, if you associate Elastic IP addresses with the channel
Frame Capture	A bucket, if Amazon S3 is set up with a VPC endpoint	Yes, if you associate Elastic IP addresses with the channel

MediaLive output type (output group)	Can be delivered to a destination in your VPC	Can be delivered to a destination outside your VPC
HLS to an HTTP or HTTPS server	A bucket, if Amazon S3 is set up with a VPC endpoint	Yes, if you associate Elastic IP addresses with the channel
HLS to an Akamai server	A bucket, if Amazon S3 is set up with a VPC endpoint	Yes, if you associate Elastic IP addresses with the channel
HLS to MediaPackage, over HTTP	No	Yes, if you associate Elastic IP addresses with the channel
HLS to MediaStore	No	Yes, if you associate Elastic IP addresses with the channel
HLS to Amazon S3	A bucket, if Amazon S3 is set up with a VPC endpoint	Yes, if you associate Elastic IP addresses with the channel
MediaPackage	No	Yes, if you associate Elastic IP addresses with the channel
Microsoft Smooth	A server on Amazon EC2	Yes, if you associate Elastic IP addresses with the channel
Multiplex	No When the channel is set up for VPC delivery, it can't contain a multiplex output.	No When the channel is set up for VPC delivery, it can't contain a multiplex output.
RTMP or RTMPS	A server on Amazon EC2	Yes, if you associate Elastic IP addresses with the channel
UDP	A server on Amazon EC2	Yes, if you associate Elastic IP addresses with the channel

Supported output codecs in AWS Elemental MediaLive

This section provides information about the output audio and video codecs that MediaLive can produce.

Topics

Codecs supported in AWS Elemental MediaLive outputs

MediaLive supports the following video codecs in outputs.

- H.264 (AVC)
- H.265 (HEVC)
- MPEG-2

MediaLive supports the following audio codecs in outputs.

- AAC
- Dolby Digital (AC3)
- Dolby Digital Plus (EAC3)
- Dolby Digital Plus with Atmos
- MPEG-1 Layer II (MP2)

Supported codecs by output type in AWS Elemental MediaLive

The following table lists the video and audio codecs that each type of MediaLive output container (output group) supports.

Container (output group)	Video codecs	Audio codecs
Archive	H.264 (AVC) H.265 (HEVC)	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3) Dolby Digital Plus with Atmos MPEG-1 Layer II (MP2)
CMAF Ingest	H.265 (AVC) H.265 (HEVC)	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3) Dolby Digital Plus with Atmos
Frame Capture	JPEG	None. A Frame Capture output doesn't include audio.
HLS with a standard container	H.264 (AVC) H.265 (HEVC)	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3) Dolby Digital Plus with Atmos
HLS with an fMP4 container	H.264 (AVC) H.265 (HEVC)	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3) Dolby Digital Plus with Atmos
MediaPackage	H.264 (AVC) H.265 (HEVC)	AAC Dolby Digital (AC3)

Container (output group)	Video codecs	Audio codecs
		Dolby Digital Plus (EAC3) Dolby Digital Plus with Atmos
Microsoft Smooth	H.264 (AVC) H.265 (HEVC)	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3)
Multiplex	H.264 (AVC) H.265 (HEVC)	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3) Dolby Digital Plus with Atmos
RTMP or RTMPS	H.264 (AVC)	AAC
UDP	H.264 (AVC) H.265 (HEVC)	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3) Dolby Digital Plus with Atmos MPEG-1 Layer II (MP2)

Supported sampling rate and bitrate for AAC output in AWS Elemental MediaLive

This section explains how to set the following four properties of the AAC audio codec:

- Profile
- Coding mode
- Sample rate

- Bitrate

In the console, these properties are in four fields in the **Codec configuration** section for the AAC codec. To get here, go to the **Create channel** page and choose the appropriate output in the output group. In **Output settings**, go to the **Audio** section. In Codec settings, choose Aac, then expand **Codec configuration**. To review the step where you complete these fields, see the section called "Set up audio".

> **Note**
>
> You can set all four fields. Or you can leave all the fields with their defaults.
> If you change only one or two fields, you might create a combination that is not valid. See the tables in the following sections to verify that the combination you have created is valid.

To set these four fields

1. Choose a **Coding mode**.
2. Choose a **Profile** that is valid with that profile. See the tables that follow this procedure.
3. Choose a **Sample rate** that is valid for that combination of profile and coding mode.
4. Choose a **Bitrate** that falls within the range that is supported for that sample rate.

Coding mode 1.0

In this table, read down the rows to find the profile that you want. Then read across to find a valid combination of sample rate and bitrate.

Profile	Sample rate (Hz)	Minimum valid bitrate (bits/sec)	Maximum valid bitrate (bits/sec)
HEv1	22050	8000	12000
	24000	8000	12000
	32000	12000	64000
	44100	18000	64000

Profile	Sample rate (Hz)	Minimum valid bitrate (bits/sec)	Maximum valid bitrate (bits/sec)
	48000	18000	64000
LC	8000	8000	14000
	12000	8000	14000
	16000	8000	28000
	22050	24000	28000
	24000	24000	28000
	32000	32000	192000
	44100	56000	256000
	48000	56000	288000
	88200	288000	288000
	96000	128000	288000

Coding mode 1+1

In this table, read down the rows to find the profile that you want. Then read across to find a valid combination of sample rate and bitrate.

Profile	Sample rate (Hz)	Minimum valid bitrate (bits/sec)	Maximum valid bitrate (bits/sec)
HEv1	32000	24000	128000
	44100	40000	192000
	48000	40000	192000
	96000	224000	256000

Profile	Sample rate (Hz)	Minimum valid bitrate (bits/sec)	Maximum valid bitrate (bits/sec)
LC	8000	16000	28000
	12000	16000	28000
	16000	16000	56000
	22050	48000	56000
	24000	48000	56000
	32000	64000	384000
	44100	112000	512000
	48000	112000	576000
	88200	256000	576000
	96000	256000	576000

Coding mode 2.0

In this table, read down the rows to find the profile that you want. Then read across to find a valid combination of sample rate and bitrate.

Profile	Sample rate (Hz)	Minimum valid bitrate (bits/sec)	Maximum valid bitrate (bits/sec)
HEv1	32000	16000	128000
	44100	16000	96000
	48000	16000	128000
	96000	96000	128000
HEv2	22050	8000	12000

Profile	Sample rate (Hz)	Minimum valid bitrate (bits/sec)	Maximum valid bitrate (bits/sec)
	24000	8000	12000
	32000	12000	64000
	44100	20000	64000
	48000	20000	64000
LC	8000	16000	20000
	12000	16000	20000
	16000	16000	32000
	22050	32000	32000
	24000	32000	32000
	32000	40000	384000
	44100	96000	512000
	48000	64000	576000
	88200	576000	576000
	96000	256000	576000

Coding mode 5.1

In this table, read down the rows to find the profile that you want. Then read across to find a valid combination of sample rate and bitrate.

Profile	Sample rate (Hz)	Minimum valid bitrate (bits/sec)	Maximum valid bitrate (bits/sec)
HEv1	32000	64000	320000

Profile	Sample rate (Hz)	Minimum valid bitrate (bits/sec)	Maximum valid bitrate (bits/sec)
	44100	64000	224000
	48000	64000	320000
	96000	240000	320000
LC	32000	160000	768000
	44100	256000	640000
	48000	256000	768000
	96000	640000	768000

Coding mode ad receiver mix

Choose this coding mode if you have an AD (audio description) audio track that you want to include in the output.

In this table, read down the rows to find the profile that you want. Then read across to find a valid combination of sample rate and bitrate.

Profile	Sample rate (Hz)	Minimum valid bitrate (bits/sec)	Maximum valid bitrate (bits/sec)
HEv1	22050	8000	12000
HEv1	24000	8000	12000
	32000	12000	64000
	44100	20000	64000
	48000	20000	64000
LC	8000	8000	14000

Profile	Sample rate (Hz)	Minimum valid bitrate (bits/sec)	Maximum valid bitrate (bits/sec)
	12000	8000	14000
	16000	8000	28000
	22050	24000	28000
	24000	24000	28000
	32000	32000	192000
	44100	56000	256000
	48000	56000	288000
	88200	288000	288000
	96000	128000	288000

Encoding schemes for video output in AWS Elemental MediaLive

Topics

Encoding schemes for the AVC (H.264) codec

The AVC (H.264) codec encoding schemes for output video include profile, bit depth, and chroma sampling. In the following table, each row is a different scheme.

Profile	Bit Depth	Chroma Sampling	Level
Baseline	8-bit	4:2:0	All levels indicated in the AVC specification
Main	8-bit	4:2:0	

Profile	Bit Depth	Chroma Sampling	Level
High	8-bit	4:2:0	
High	10-bit	4:2:0	
High	8-bit	4:2:2	
High	10-bit	4:2:2	

Encoding schemes for the HEVC (H.265) codec

The HEVC (H.265) codec encoding schemes for output video include profile, bit depth, chroma sampling, tier, and level. In the following table, each row is a different scheme.

Profile	Bit Depth	Chroma Sampling	Tier	Level
Main	8-bit	4:2:0	Main	All levels indicated in the HEVC specification
Main	8-bit	4:2:0	High	
Main	10-bit	4:2:0	Main	
Main	10-bit	4:2:0	High	

Encoding schemes for MPEG2

The MPEG2 codec encoding schemes for output video include profile, bit depth, and chroma sampling.

Profile	Bit Depth	Chroma Sampling
Main	8-bit	4:2:0

Supported resolutions for video outputs in AWS Elemental MediaLive

In the following table, each row defines the video resolutions that apply to the terms SD, HD, and UHD. The table also specifies the resolutions that are supported with each codec.

Resolution	Definition	Supported in AVC outputs	Supported in HEVC outputs	Supported in MPEG2 codec
SD	Vertical resolution under 720	Yes	Yes	Yes
HD	Vertical resolution over 720, up to and including 1080	Yes	Yes	
UHD or 4K	Vertical resolution over 1080, up to and including 2160	Yes	Yes	

Identifiers for variable data

Identifiers for variable data are $ codes that you can include in a field value to represent variable data. Typically, MediaLive resolves the variable data (for example, `$dt$` for the date and time) when you run the channel. For example, `$dt$` resolves to the current date and time.

When you use these identifiers, make sure that the channel doesn't end up with two (or more) outputs with identical destinations. If that happens, the channel passes validation upon creation, but fails on start.

The following sections describe the variable identifiers that MediaLive supports, and the rules for where you can use these identifiers.

Supported variable data

MediaLive supports the variable data identifiers listed in the following table. In each row, the first column specifies the string to enter in a field. The second column specifies the format of the data after MediaLive has resolved the variable. The third column describes the data.

Identifier	Format	Description
`$dt$`	YYYYMMDDTHHMMSS	For HLS outputs, the UTC date and time of each segment. For all other outputs, the UTC date and start time of the channel.
`$d$`	YYYYMMDD	For HLS outputs, the UTC date and time of each segment. For all other outputs, the UTC date when the channel starts.
`$t$`	HHMMSS	For HLS outputs, the UTC time of each segment. For all other outputs, the UTC start time of the channel.
`$rv$`	Kb	Video bitrate.
`$ra$`	Kb	Total of all audio bitrates in the output.
`$rc$`	Kb	Container bitrate for the output, or the sum of video and all audio bitrates for the output, if the container bitrate is not specified.

Identifier	Format	Description
w	Pixels	Horizontal resolution.
h	Pixels	Vertical resolution.
f	Integer	FPS frame rate without decimal places. For example, “23.976” appears as “23”.
$$	$	Escaped $.
sn	Integer, fixed length	Number of the segment of the video in the output.
%0n	Padding modifier	Modifier for any data identifie r. The modifier pads the resolved value with leading zeros. The format is%0n, where *n* is a number. For example, to ensure the resolved value in the h identifier is 5 characters long, specify the identifier as $h %05$. If the vertical resolution is “720”, then the resolved, padded value is “00720”.

Rules for using variable data

This table describes where you can use the variable data identifiers from the previous table. In each row, the first two columns specify where you can use identifiers. The third column specifies which identifiers you can use in that location.

Object	Field	Acceptable Identifiers
Channel – Archive, HLS, Microsoft Smooth output groups	Destination field in an Output group	`$dt$`, `$d$`, `$t$`
Channel – Archive, Microsoft Smooth output groups	Name modifier field in an Output	All except `$ra$`, `$rc$`, `$sn$`
Channel – HLS output groups	Name modifier field in an Output	All except `$sn$`
Channel – Archive, Microsoft Smooth output groups	Segment modifier field in an Output	All except `$ra$`, `$rc$`, `$sn$`
Channel – HLS output groups	Segment modifier field in an Output	All except `$sn$`
Schedule – HLS ID3 Segment Tagging action	Tag field	All

Features of AWS Elemental MediaLive

This chapter contains detailed procedures for implementing AWS Elemental MediaLive features. You set up these features when you create or modify the channel or when you add actions to the channel schedule. The procedures expand on the limited information provided in *Setup: Creating a channel* and *Setup: Creating a schedule*.

Topics

- Working with metadata in AWS Elemental MediaLive
- Working with motion graphics overlays
- Multiplex and MPTS in AWS Elemental MediaLive
- Nielsen watermarks
- Converting Nielsen watermarks to ID3
- Implementing pipeline locking
- Implementing pipeline redundancy
- Implementing resiliency in the channel
- SCTE-35 message processing
- Sharing encodes among outputs
- Handling SMPTE 2038 metadata
- Using ACLs for delivery to Amazon Simple Storage Service
- Tagging AWS Elemental MediaLive resources
- Viewing input thumbnails in AWS Elemental MediaLive
- Timecodes
- Implementing a trick-play track
- Handling a straightforward color space conversion
- Handling complex color space conversions
- Setting up enhanced VQ mode
- Setting the rate control mode
- Delivering outputs via your VPC

Including audio accessibility data in AWS Elemental MediaLive

In the audio in CMAF Ingest or Microsoft Smooth output groups, you can include accessibility data. This data describes the type of accessibility that the encode represents. For example, an audio track might actually be a spoken description of what's happening in the video. Accessibility data is also known as accessibility signaling.

MediaLive also includes a feature for including accessibility data in captions. For more information, see the section called “Including accessibility data in captions”.

Supported accessibility data standards

MediaLive supports the following styles of accessibility data.

Accessibility data style	Specification	CMAF Ingest	Microsoft Smooth
DASH role audio	DASH role scheme (*ISO/IEC 23009-1:2022(E)*)	Yes	Yes
DVB DASH accessibility	*ETSI TS 103 285 Technical Specification, V1.3.1 (2020-02)*	Yes	Yes

Specifying accessibility data

> **Note**
>
> The information in this section assumes that you are familiar with the general steps for creating or editing a channel.

1. In the **Create channel** or **Edit channel** page for the channel, in the **Channel** panel, find the output group that you want to set up. Then find the audio output where you want to configure accessibility data.
2. Select the output by its name. The details appear on the right. Go to the **Stream** settings section and choose the **Audio** section.
3. Open the **Additional settings** and set the accessibility data fields.
   - To include DASH Roles, choose **Add dash roles** as many times as you want. In **DASH Role Audio**, choose the style in each role.
   - To include DVB DASH accessibility style, in **DVB DASH accessibility**, choose the applicable description. You can add only one instance of this accessibility style.

You can add more than one style of acessibility data to each encode. For example, you can add Dash Roles and DVB DASH accessibility style. You might want to do this because different downstream systems for these outputs implement different styles.

Handling of accessibility data

The fields for accessibility data appear in the encode fields in all output group types, including types that don't support this data.

> **Note**
>
> When you set up audio encodes and you plan to include accessibility data, proceed as follows. First create the audio encodes in the CMAF Ingest and/or Microsoft Smooth output groups, and set up the accessibility data. Then create the audio encodes in the other output groups.

Handling in supported output groups

If you aren't implementing shared audio encodes, MediaLive includes the data only in the audio outputs of the CMAF Ingest and Microsoft Smooth output groups that you set up for audio accessibility data.

Handling in shared encodes

You might plan to share audio encodes among several output groups. For example, you might share an audio encode among at least one CMAF Ingest or Microsoft output group, and with other output groups.

If you set up accessibility data in a shared audio encode, MediaLive will handle the data as follows:

- It will include the data in the CMAF Ingest and Microsoft Smooth output groups that share the encode.
- It won't include the data in other output groups, because those output groups don't support this data. Even though the output group is sharing the encode, MediaLive won't include the data.

Handling in other output groups

You might try to set up accessibility fields in an output that doesn't support accessibility data. If you're not implementing encode sharing with a CMAF Ingest or Microsoft Smooth output group, you will get an error message when you save the channel.

Audio-only outputs

You can set up a MediaLive channel with an output group that contains only audio. You can create a channel with these combinations of output groups:

- One audio-only output group.
- Several audio-only output groups.
- One or more audio-only output groups and other regular (video-plus-audio) output groups.

Set up the channel in the usual way, using the guidelines in this section to set up the inputs, outputs, and encodes for the audio-only output groups.

> **Note**
>
> The information in this section assumes that you are familiar with the general steps for creating a channel, as described in *Setup: Creating a channel* .

Topics

Inputs

Setting up the input source

The channel can have a single input or multiples inputs. All the output groups (both those that are audio-only and those that are video-and-audio) always ingest the same inputs.

You can use one of the following two categories of input.

- Input that contains *only audio*. In this case, the input must be one of these inputs and must be in a Transport Stream (TS) container:
  - MediaConnect input
  - RTP input
- Input that contains *both audio and video* (and optionally captions). In this case, the input can be any input type that MediaLive supports.

Setting up input attachments

In each input attachment, create as many audio selectors as you require. For example, create a selector for each language to extract. Or create a selector for each audio quality or codec that is available.

Keep in mind that in a channel with both audio-only and audio-and-video output groups, you don't have to create special audio selectors for the sole use of the audio-only output. The same audio selector can be used by both audio-only and audio-and-video output groups.

Output groups and outputs

Setting up output groups

You can create an audio-only output in the following types of output groups.

- HLS
- Microsoft Smooth
- RTMP
- UDP

Setting up outputs

The following list describes the number and type of required outputs, based on the output group.

HLS output group

Create outputs in the output group as follows:

- If the output group contains *only one audio encode*, then create one output. Set the container type to **Audio-only**.

- If the output group contains *more than one audio encode*, then set up an audio rendition group that doesn't include video. See the section called “Audio – audio rendition groups for HLS”.

Microsoft Smooth output group

Create one output for each audio encode.

RTMP output groups

Create one output for the single audio encode. (RTMP always supports only one audio in each output group.)

UDP output groups

Create one output for all the audio encodes.

Streams

Setting up encodes in streams

Use the following settings for the streams in the output.

All outputs *except UDP*

In the **Streams settings** section for each output, set up so that each output has one and only one audio encode. Therefore you must do the following.

- Remove the video encode that MediaLive automatically adds.
- Make sure that you don't add any captions encodes.

UDP outputs

In the **Streams settings** section for the single output, set up so that each output contains only audio encodes. Therefore you must do the following.

- Add as many audio encodes as you require.
- Remove the video encode that MediaLive automatically adds.
- Make sure that you don't add any captions encodes.

Configuring encodes

In the **Streams settings** section, in **Audio**, set up each encode as follows.

- In **Audio selector name**, choose one of the audio sources that you set up when you configured the input attachment.
- In **Codec settings**, choose any output audio codec that the output type supports.

Dolby Digital Plus with Dolby Atmos

Dolby Atmos is a surround sound technology built on the codec of Dolby Digital Plus (Dolby Digital Plus is also known as Enhanced AC-3). In addition to the multi-channel surround sound capabilities of Dolby Digital Plus, Dolby Atmos adds additional height-based audio channels.

The following handling is supported:

- Encoding an audio output as Dolby Digital Plus with Dolby Atmos. The audio input must be a source that contains up to 16 channels.
- Passthrough of input audio that is already Dolby Digital Plus with Dolby Atmos.

> **Note**
>
> MediaLive doesn't support decoding of Dolby Digital Plus with Dolby Atmos.

Supported inputs

The input must have these characteristics:

- To encode Dolby Digital Plus with Dolby Atmos, the audio input can be any audio source that MediaLive supports and that has these characteristics:
  - Up to 16 channels in the following order:

    ```
    L R C LFE Ls Rs Lb Rb Tfl Tfr Tsl Tsr Tbl Tbr Lw Rw
    ```

  - If the source has fewer than 16 channels, MediaLive extracts all the channels and then pads the output by inserting silence in the higher-numbered channels. For example, if the source

has two channels, MediaLive puts those channels in L and R, then inserts silence in the remaining channels.

- If the source doesn't have the channels in the specified order, the results might be wrong on the downstream player. For example, the sound of rain falling might come out of the left speaker instead of a ceiling speaker.
- A sampling rate of 48000 Hz.

- To pass through a Dolby Digital Plus with Dolby Atmos source, the audio can be any coding mode and any sampling rate that Dolby Digital Plus supports.

Supported outputs

Audio encoding

The MediaLive implementation of Dolby Digital Plus with Dolby Atmos supports the following coding modes in the output:

- 5.1.4 coding mode
- 7.1.4 coding mode
- 9.1.6 coding mode

Within each coding mode, the speaker channels are arranged as shown in the following table.

Coding mode	Channel arrangement
5.1.4	`L R C LFE Ls Rs Tfl Tfr Tbl Tbr`
7.1.4	`L R C LFE Ls Rs Lb Rb Tfl Tfr Tbl Tbr`
9.1.6	`L R C LFE Ls Rs Lb Rb Tfl Tfr Tsl Tsr Tbl Tbr Lw Rw`

The abbreviations are the standard Dolby abbreviations: Left, Right, Center, LFE (Low Frequency Effects), Left surround, Right surround, Left back, Right back, Top front left, Top front right, Top side left, Top side right, Top back left, Top back right, Left wide, and Right wide.

Setting up the channel

Follow this procedure to produce Dolby Digital Plus with Dolby Atmos in one or more outputs.

> **Note**
>
> The information in this section assumes that you are familiar with the general steps for creating a channel.

To set up the input

Follow this procedure if the source audio is Dolby Digital Plus, to convert the audio to Dolby Digital Plus with Dolby Atmos.

1. In the channel in MediaLive, select the input that contains the Dolby Digital Plus audio that you want to transcode or pass through.
2. In the **General input settings** section choose **Add audio selectors**.
3. Complete the fields to extract the Dolby Digital Plus audio.

To set up the output if the source audio is Dolby Digital Plus

1. In the channel, go to the output group where you want to add the audio. Or create a new group.
2. Create the output where you want to add the audio encode.
3. In the **Stream settings** section for the output, choose the **Audio** section. Complete the fields as follows.

Field	Description
Audio Selector Name	Choose the audio selector that you set up in the input.
Codec Settings	Choose **EAC3 ATMOS**.
Bitrate	Choose a value that is applicable to the coding mode.

Field	Description
Coding mode	Choose the coding mode you want. For more information, see the section called "Supported outputs".
Dialnorm	Choose the dialogue normalization value. The dialnorm setting adjusts the output audio gain. Dialnorm is an integer value with a range of 1 to 31 corresponding to an output audio gain of -30 to 0 dB, respectiv ely.
DRC Line	Choose a Dolby dynamic range compressi on profile. For more information about how each profile handles compression, see the Dynamic Range Control section of Dolby's Metadata guide for developers: A Guide to Dolby Metadata.
DRC RF	
Surround Trim	Choose the maximum amount of attenuati on to apply to Surround and Height channels when the downstream player isn't configured to handle Dolby Digital Plus with Dolby Atmos and must remix the channels.
Height Trim	

4. Complete the additional audio fields as desired.

To pass through Dolby Digital Plus with Dolby Atmos from the input to the output

Follow this procedure if the source audio is already Dolby Digital Plus with Dolby Atmos.

1. In the channel, go to the output group where you want to add the audio. Or create a new group.
2. Create the output where you want to add the audio encode.
3. In the **Stream settings** section for the output, choose the **Audio** section.
4. Set these fields:

- **Audio Selector Name**: Set to the audio selector that you set up in the input.
- **Audio Codec**: Set to **Passthrough**.

With this setup, the selected audio source in the input will be passed through.

> **Important**
>
> Don't set **Audio Codec** to **EAC3 ATMOS**. That isn't the correct value for passing through. If you choose this option, the output might have silent audio.

Sample HLS manifest

If you include Dolby Digital Plus with Dolby Atmos in an HLS output group, the audio line in the HLS manifest looks like this example:

```
#EXTM3U
#EXT-X-VERSION:4
#EXT-X-INDEPENDENT-SEGMENTS
#EXT-X-STREAM-INF:BANDWIDTH=2208800,AVERAGE-
BANDWIDTH=2142800,CODECS="avc1.64001f,ec-3",RESOLUTION=1280x720,FRAME-
RATE=30.000,AUDIO="program_audio_0"
index_video.m3u8
#EXT-X-
MEDIA:TYPE=AUDIO,LANGUAGE="eng",NAME="English",AUTOSELECT=YES,DEFAULT=YES,CHANNELS="12/
JOC",GROUP-ID="program_audio_0",URI="index_audio.m3u8"
```

The `Channels` attribute in the last line is significant for Dolby Digital Plus with Dolby Atmos:

- 12/JOC indicates that the coding mode is 5.1.4 or 7.1.4 and the codec is Dolby Digital with Dolby Atmos.
- 16/JOC indicates that the coding mode is 9.1.6 and the codec is Dolby Digital with Dolby Atmos.

Dolby E input audio in MediaLive

You can set up AWS Elemental MediaLive to ingest Dolby E audio. The Dolby E audio must be wrapped in a PCM streams tagged with SMPTE-337. The options for handling this audio source are the following:

- Extract individual programs from the source audio, and then convert it or remix it in the output.
- Pass them through all the programs, with no conversion and no remixing.
- Set up the source for both types of handling—to extract programs and to pass through the entire source.

Topics

About Dolby E

Dolby E wrapped in PCM can carry up to eight *Dolby E programs* delivered in two *audio tracks*. The two audio tracks are a standard stereo pair (in other words, 2.0 coding mode).

The two tracks contain the number of Dolby E programs required for the coding mode of the audio. For example, if the audio is 7.1 audio, then all the Dolby E programs contains content. If the audio is four stereo languages, then all the Dolby E programs contain content. But if the audio is only three stereo languages, then only six of the Dolby E programs contain content.

The supported coding modes for Dolby E audio are AD, 1.0 (mono), 1.1, 2.0 (stereo), 3.2, 4.0, 5.1, and 7.1. All these coding modes are supported by MediaLive on the input side.

For information about the input types that support Dolby E, see the section called “Input codecs”.

Getting ready

Contact the content provider for this input to find out about the programs that are included in the Dolby E audio. For example, it might contain 7.1 audio in English. Or it might contain four sets

of stereo (English, French, Spanish, Punjabi) with English in Dolby E programs 1 and 2, and so on, through to Punjabi in Dolby E programs 7 and 8.

Setting up the input to extract programs

You must identify each Dolby E program that you want to extract and map it to a *MediaLive audio selector*. Each MediaLive audio selector maps to one Dolby E program.

> **Note**
>
> The information in this section assumes that you are familiar with the general steps for creating a channel.

1. In the channel in MediaLive, select the **Input attachment** that contains the Dolby E audio that you want to decode or pass through.
2. In the **General input settings** section choose **Add audio selectors**. Fields for one audio selector (**Audio Selectors 1**) appear.
3. Complete the fields as follows.

Field	Description
Audio selector name	Assign a name to the Dolby E program. For example, **`DolbyE program1`**.
Selector settings	From the drop-down menu, choose **Audio track selection**.
Dolby E decode	Choose **Audio Dolby E decode**. The **Dolby E program selection** field appears. The drop-down menu shows the eight possible Dolby E programs.
Dolby E program selection	Select the Dolby E program that you want to extract. For example, **PROGRAM_1**.

Don't select the Add tracks field. This field doesn't apply to Dolby E audio.

4. To extract more Dolby E programs, choose **Add audio selectors** as many times as you need. Follow the steps above for each Dolby E program that you want to extract.

When you have finished, there will be one audio selector for each program to extract.

Setting up the input to pass through the audio

You can pass through the entire Dolby E audio source so that you can then pass it through in the output.

> **Note**
>
> The information in this section assumes that you are familiar with the general steps for creating a channel.

1. In the channel in MediaLive, select the **Input attachment** that contains the Dolby E audio that you want to decode or pass through.
2. In the **General input settings** section choose **Add audio selectors**. Fields for one audio selector (**Audio Selectors 1**) appear.
3. Complete the fields as follows.

Field	Description
Audio selector name	Assign a name to the Dolby E program. For example, **`DolbyE passthrough`** .
Selector settings	From the drop-down menu, choose **Audio track selection**.
Dolby E decode	Choose **Audio Dolby E decode**. The **Dolby E program selection** field appears. The drop-down menu shows the eight possible Dolby E programs.
Dolby E program selection	Select the Dolby E program that you want to extract. For example, **ALL_CHANNELS**.

Setting up the input to extract and pass through

You can set up the source in both ways—to extract programs and to pass through the entire source.

In the same input attachment, set up one selector for passthrough, and set up several selectors to extract programs.

Audio rendition groups for HLS

You can set up an HLS output group to include an audio rendition group. An audio rendition group is a set of MediaLive audio encodes (for example, a set of languages) that is associated with a video. Audio rendition groups let the downstream client player select a video and then select from among several audio encodes that all apply to that video.

Each audio encode in an audio rendition group is called an *audio rendition* or an *audio variant* or an *audio variant stream*.

You can set up the HLS output group in one of these ways:

- As a regular HLS output group, with video, audio (in a rendition group), and optional captions.

  The video might be associated with only one audio rendition group, or it might be associated with several. For example, the video might be associated with one group consisting of high-bitrate audio and another group consisting of low-bitrate audio.

  Or one audio rendition group might be associated with several videos. For example, the same audio rendition group might be associated with the high, medium, and low-bitrate video offerings.
- As an audio-only rendition group.

  In this case, follow the steps in this procedure, but ignore steps for setting up video.

> **Note**
>
> The information in this section assumes that you are familiar with the general steps for creating a channel, as described in *Setup: Creating a channel* .

The key fields in the console that relate to this feature are under the **HLS Settings** field on the **Output settings** section of the **HLS output group** section on the **Create channel** page. To review the step where you complete these fields, see the section called “The procedure”.

Topics

About audio rendition groups

Standards compliance

This implementation of audio rendition groups is compliant with *HTTP Live Streaming draft-pantos-http-live-streaming-18* section 4.3.4.1.1.

Examples

Example 1

The HLS output group consists of:

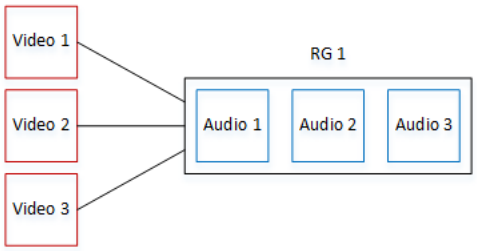
- One video output.
- Three audio outputs (perhaps English, French, Spanish) that all belong to the same audio rendition group.

Example 2

The HLS output group consists of:

- One *video high* output.
- One *video medium* output.

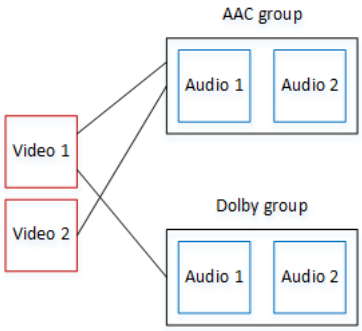
- One *video low* output.
- Three audio outputs (English, French, Spanish) that all belong to the same audio rendition group.



Example 3

The HLS output group consists of:

- One *video high* output.
- One *video low* output.
- Two audio outputs (English, French) that each use the AAC codec. These outputs both belong to the same audio rendition group, RG1.
- Two audio outputs (English, French) that each use the Dolby Digital codec. These outputs both belong to the same audio rendition group, RG2.
- The video high output is associated with both audio rendition groups.
- The video low output is associated only with the RG1 audio rendition group.



Creating an output with an audio rendition group

This section describes how to create audio rendition groups in an HLS output group and how to associate those groups with the appropriate video outputs (if any). The encodes and associations that you create are the following:

- If you want to include video in the output group, then for each video asset, you create one video output containing one video encode. The output can also contain embedded captions, but it can't include sidecar captions. The output can't contain audio encodes.
- For each audio asset, you create one *audio-only* output containing one audio encode and no other encodes.
- You decide on an ID for each rendition group. The ID is a name that you decide on. For example *AAC audio group*.
- To group several audio outputs into one rendition group, you assign the same *audio group ID* to each audio output.
- Finally, to associate the video output (if any) with the audio rendition group, you assign the *audio group ID* to that video output.

Topics

Step 1: Identify the video and audio encodes

You must plan the requirements for the audio rendition group. You must identify the video encodes that you want in the output group. You then decide on the individual audio encodes. Finally, you identify the audio rendition groups you want each encode to belong to.

To identify and map the encodes

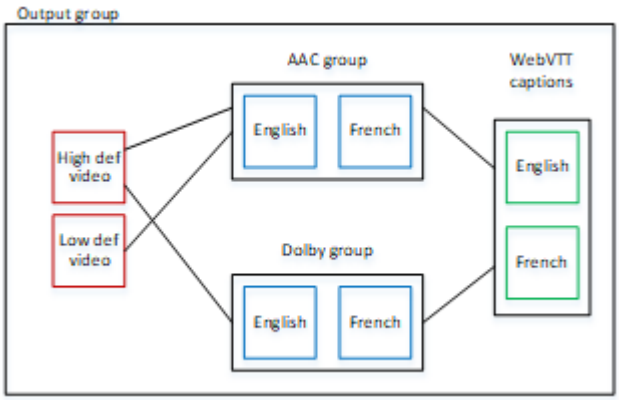
1. Identify any video encodes that you require in the HLS output group. For example, one high-resolution encode and one low-resolution encode.
2. Identify the audio encodes that you require. For example, AAC in English and French, and Dolby Digital in English and French.
3. Decide how many audio renditions you require. Review the rules to ensure that you design a rendition group that is valid.
4. Give a name to each video, audio, and audio rendition group. For example:

- A video output named **`high definition`**.
- A video output named **`low definition`**.
- Audio English AAC named **`AAC EN`**.
- Audio French AAC named **`AAC FR`**.
- Audio English Dolby Digital named **`DD EN`**.
- Audio French Dolby Digital named **`DD FR`**.
- A rendition group named **`AAC group`** for AAC audio.
- A rendition group named **`DD group`** for Dolby Digital audio.

5. Identify how you want the video to be associated with the audio rendition groups. For example:

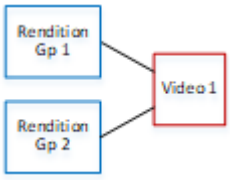
   - Video **`high definition`** to be associated with **`AAC group`** and **`DD group`**.
   - Video **`low definition`** to be associated only with **`AAC group`**.

6. (Optional) For completeness in designing the output group, identify the captions that you require.

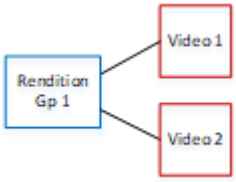


Rules for video and audio in rendition groups

- Both video and captions are optional.
- A video encode can be associated with more than one rendition group. For example, *video high* can be associated with both *Dolby audio* and *AAC audio*. There is no need to create separate video encodes for each rendition group.



- All the rendition groups associated with the same video must contain the same audio encodes. For example, if both the AAC group and the Dolby group are associated with the high definition video encode, both these groups must contain the same audio languages (perhaps English, French, and Spanish).
- An audio encode can belong to only one audio rendition group.
- An audio rendition group can be associated with more than one video. For example, the Dolby group can be associated with the high definition video encode and the low definition video encode. There is no need to create separate rendition groups for each video.



Step 2: Determine Defaults and Selection Rules

As the second part of planning the audio rendition group, you should identify the following:

- The rendition (if any) that is the default.
- How auto-selection will work for the non-default renditions.

This information might be useful to the client player that is playing this media asset.

- If a client player is configured with an audio preference (for example, Spanish) and that preference is not available, the player can use this information to select an audio.
- Or if the client player is not configured with any audio preference, the client player can use this information to select an audio.

(If the preference that is configured in the client player is available, the player ignores this information and selects that preference.)

To determine defaults and auto-selection behavior

- For each audio rendition in the rendition group, choose the behavior from the following table. Each audio can have a different value.

  Each row in the following table describes a different behavior.

Value for a given audio rendition	Client player behavior	Representation in HLS Manifest
Alternate Audio, Auto Select, Default	The client player should select this audio rendition . Only one audio rendition s in the rendition group should be set as the default, otherwise the client player might behave unexpectedly.	`EXT-X-MEDIA` with `DEFAULT=YES` , `AUTOSELECT=YES`
Alternate Audio, Auto Select, Not Default	The client player might select this audio rendition. Any number of renditions in the rendition group can be set this way.	`EXT-X-MEDIA` with `DEFAULT=NO` , `AUTOSELECT=YES`
Alternate Audio, not Auto Select	The client player should never select this audio rendition. Any number of renditions in the rendition group can be set this way.	`EXT-X-MEDIA` with `DEFAULT=NO` , `AUTOSELECT=NO`
Audio-Only Variant Stream	The client can play back this audio-only rendition instead of video, in low-bandwidth scenarios.	`EXT-X-STREAM-INF`

Example 1

In this example you want to set up the audio rendition group so that the client player can auto-select any of the renditions. You also want a default audio in the rendition group in case the client player is not set up with a default.

- Set only one audio rendition to *Alternate Audio, Auto Select, Default*.
- Set every other audio rendition to *Alternate Audio, Auto Select, Not Default*.
- Optionally, if you have an audio rendition that plays when the bandwidth is so low that the video cannot be delivered, then set that audio rendition to *Audio-Only Variant Stream*.

Example 2

In this example you want to set up the audio rendition group so that the client player can auto-select only specific renditions. You also want a default audio in the rendition group in case the client player is not set up with a default.

- Set only one audio rendition to *Alternate Audio, Auto Select, Default*.
- Set some of the other renditions to *Alternate Audio, Auto Select, Not Default*.
- Set some of the other renditions to *Alternate Audio, not Auto Select*.
- Optionally, if you have an audio rendition that plays when the bandwidth is so low that the video cannot be delivered, then set that audio rendition to *Audio-Only Variant Stream*.

Example 3

In this example you want to set up the audio rendition group so that the client player can auto-select any audio rendition it chooses. You don't want a default audio rendition in the rendition group, so the client player always auto-selects audio.

- Set every audio rendition to *Alternate Audio, Auto Select, Not Default*.
- Optionally, if you have an audio rendition that plays when the bandwidth is so low that the video cannot be delivered, then set that audio rendition to *Audio-Only Variant Stream*.

Step 3: Create the Video Outputs

If you want to include video in the HLS output group, then follow this procedure for each video encode.

This procedure involves the following fields in the output section of the HLS output group the Channel page in the console:

- **Output settings** – **HLS settings**
- **Output settings** – **HLS settings** – **Audio rendition sets**

To create the video outputs

1. In the **HLS output group**, in **HLS outputs**, choose **Add output**. Choose **Settings** to display the **Outputs** page for that output.
2. In the **Output** page, set up as follows:
   - In **Stream settings**, choose **Audio 1** and choose **Remove audio**. The output now contains only a video encode. In an output group that includes audio rendition groups, each video encode must each be in a video-only output.
   - Set up the rest of the stream settings as described in Fields for the video, audio, and captions streams (encodes).
3. Complete the **Output settings** section as follows:
   - For **HLS settings**, choose **Standard hls** or **Fmp4**, as appropriate.
   - For **Audio rendition sets**, enter the name of the audio renditions groups to associate with this video output. Don't worry that you haven't created this name yet. To associate the video output with more than one group, enter a comma-separated list. For example:

     ```
     AAC group, DD group
     ```

Step 4: Create the audio outputs

Follow this procedure for each audio encode in the HLS output group.

This procedure involves the following fields in the output section of the HLS output group the Channel page in the console:

- **Output settings** – **HLS settings**
- **Output settings** – **HLS settings** – **Audio track type**
- **Output settings** – **HLS settings** – **Audio group ID**
- **Output settings** – **HLS settings** – **Segment type**

To create each audio output

1. In the **HLS output group**, in **HLS outputs**, choose **Add output**. Choose **Settings** to display the **Outputs** page for that output.
2. In the **Output** page, set up as follows:
   - In **Stream settings**, choose **Video 1** and choose **Remove video**. The output now contains only an audio encode. In an output group that includes audio rendition groups, each audio encode must each be in its own output.
   - Set up the rest of the stream settings in the usual way.
3. Complete the **Output settings** section as follows:
   - For **HLS settings**, choose **Audio only hls**. More fields appear.
   - For **Audio track type**, choose the value you decided on for this audio encode when you determined the default.
   - For **Audio group ID**, enter the name of the rendition group that you want this audio encode to belong to. For example, enter **`AAC group`**. One audio encode can belong to only one rendition group.
   - For **Segment type**, choose **AAC**.

   Ignore **Audio only image**. This field does not apply to audio rendition groups.

Summary

After following these steps, you have the following outputs:

- Two or more audio-only outputs. Each output belongs to the audio rendition group specified in **Audio Group ID**.
- Optionally, one or more video outputs. Each output is associated through the audio rendition group or groups specified in **Audio Rendition Sets**.

Sample manifest

This sample manifest contains the following elements:

- Two video outputs, as indicated by the presence of two EXT-STREAM-INF lines (the last two lines in the example).
  - The first video output has a low bandwidth. As indicated by the AUDIO parameter, it is associated with *audio1*.
  - The second video output has a higher bandwidth. As indicated by the AUDIO parameter, it is associated with *audio2*.
- Four audio outputs, as indicated by the presence of four EXT-X-MEDIA lines with TYPE=AUDIO. In each audio output, the values for the parameters come from the following fields in each audio output in the channel:
  - Type is always Audio.
  - GROUP-ID is from the **Audio Group ID** field in the **Output settings** section.
  - LANGUAGE is from the **Language Code** field in the **Stream settings** section.
  - NAME is from the **Stream Name** field in the **Stream settings** section.
  - AUTOSELECT and DEFAULT are from the **Alternate Audio Track Type** field in the **Output settings** section.
  - URI is from the **Destination** field in the output group.

  For information on all these fields, see the section called “Step 4: Create audio”.
- Two audio rendition groups. The audio renditions groups don't have their own lines in the manifest. Their existence is implied by the presence of GROUP-ID parameters in the audio lines.
- Two captions streams, as indicated by the presence of two EXT-X-MEDIA lines with TYPE=SUBTITLES.

```
#EXTM3U
#EXT-X-MEDIA:TYPE=AUDIO,GROUP-ID="AAC
 group",LANGUAGE="eng",NAME="English",AUTOSELECT=YES,\ DEFAULT=YES,URI="eng1/aac-
en.m3u8"
#EXT-X-MEDIA:TYPE=AUDIO,GROUP-ID="AAC
 group",LANGUAGE="fre",NAME="français",AUTOSELECT=YES,\ DEFAULT=NO,URI="fr1/aac-
fr.m3u8"
#EXT-X-MEDIA:TYPE=AUDIO,GROUP-ID="DD
 group",LANGUAGE="eng",NAME="English",AUTOSELECT=YES,\ DEFAULT=YES,URI="eng2/dd-
en.m3u8"
#EXT-X-MEDIA:TYPE=AUDIO,GROUP-ID="DD
 group",LANGUAGE="fr",NAME="français",AUTOSELECT=YES,\ DEFAULT=NO,URI="fr2/dd-fr.m3u8"
```

```
#EXT-X-MEDIA:TYPE=SUBTITLES,GROUP-ID="subs",LANGUAGE="eng",NAME="English",
DEFAULT=YES,AUTOSELECT=YES,FORCED=NO,URI="sub-en.m3u8"
#EXT-X-MEDIA:TYPE=SUBTITLES,GROUP-ID="subs",LANGUAGE="fra",NAME="French",
DEFAULT=YES,AUTOSELECT=YES,FORCED=NO,URI="sub-fr.m3u8"

#EXT-X-STREAM-INF:PROGRAM-
ID=1,BANDWIDTH=195023,CODECS="avc1.42e00a,mp4a.40.2",AUDIO="AAC group"
lo/prog-index.m3u8,SUBTITLES="subs",URI="curling-hi.m3u8"
#EXT-X-STREAM-INF:PROGRAM-
ID=1,BANDWIDTH=591680,CODECS="avc1.42e01e,mp4a.40.2",AUDIO="DD group"
hi/prog-index.m3u8,URI="curling-lo.m3u8"
```

Working with AWS Elemental Link

Read this section if your organization uses AWS Elemental Link hardware devices as video sources for inputs with AWS Elemental MediaLive channels. (Your organization might also use AWS Elemental Link hardware devices as video sources for MediaConnect flows. For information about that usage, see the section called “Using Link with a MediaConnect flow”.)

AWS Elemental Link is a *hardware device* that connects a live video source, such as a camera or video production equipment, to MediaLive. The AWS Elemental Link hardware device connects to AWS over a secure connection that AWS manages. For information about purchasing AWS Elemental Link, see Elemental Appliances and Software.

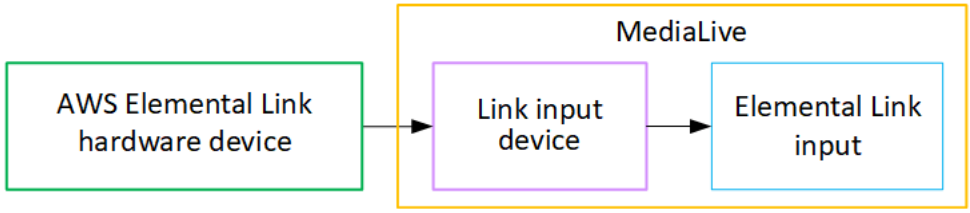
There are two versions of the device:

- AWS Elemental Link HD, which can handle HD sources. This is the *HD device*.
- AWS Elemental Link UHD, which can handle HD and UHD sources. This is the *UHD device*.

For more information about the versions of the devices, see the section called “HD and UHD devices”.

After the hardware device is connected, it automatically appears in MediaLive as a *Link input device*. The *Link input device* is an interface in MediaLive for the external hardware device. In other words, you use the *Link input device* in MediaLive to work with the external hardware device.

After the Link input device exists, you create an *Elemental Link input* that uses that Link input device. You can then use the input as you would use any input—you attach the input to a channel.



To clarify the terminology:

- AWS Elemental Link (AWS Elemental Link) is a *physical hardware device*.
- Link input device is the *interface* for AWS Elemental Link in MediaLive. It is the term that you see in the console.
- Elemental Link input is a *type of input* in MediaLive.

Topics

Using AWS Elemental Linkfor a MediaLive input

You can set up an HD device or a UHD device as the source for a MediaLive input. You can then attach the input to a MediaLive channel.

Set up the device

Follow this procedure if the Link device is new to your organization,

1. Set up the device on the internet. For more information, see the section called “Deploying the hardware”.
2. Ask your IAM administrator to give you IAM permissions to work with the Link input device interface. See the section called “Setting up users with IAM permissions”.
3. Sign in to the AWS Management Console and open the MediaLive console at https://console.aws.amazon.com/medialive/.
4. Find out if your organization obtained your device from an AWS reseller. If so, you must claim it.

Set up the Link input device

Use MediaLive to perform these steps.

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Input devices**. The devices that you have access to appear in the **Device list** page. Find the device that you want. If you can't find it, follow the troubleshooting tips in the section called “Viewing details about devices”.
3. Get set up in the correct Region. The device and the flow must be in the same Region. Follow these steps to get aligned:
   - Decide on the Region where you will work.
   - If you want to work in a different Region, transfer the device now. Then switch the MediaLive console to that Region. From now on, make sure that you work in this Region.
4. When the device appears in the **Device list** page, choose the link on the individual card to display the **Device details** page.
5. Check the message on the **Attachments** tab to determine how the device is currently being used.

   Device is not being used

   A message specifies that the device isn't being used, which means that it isn't connected to a MediaLive input or a MediaConnect flow.

   In this case, the device is ready for you to set it up. See the next procedure.

   Device is being used for inputs

   A message specifies that the device is already being used as an input source. You can set up the device as the source for another input, to a maximum of four inputs for a device. Make a note of the current Region. You will have to use the device in the existing Region.

   In this case, the device is ready for you to set it up. See the next procedure.

   Device is being used for a flow

   A message specifies that the device is already being used as a source for a MediaConnect flow.

   To use this device for a flow, you must first decommission the current usage. You should check with other people in your organization to confirm that no one else plans to use this device is its current usage. Then choose **Detach MediaConnect flow**.

   After the card clears, the device is ready for you to set it up. See the next procedure.

Configure the device

Use MediaLive to set up the device.

1. Get set up in the correct Region. The device, the input, and the channel must be in the same Region, and you must work in that Region. Follow these steps to get aligned:
   - Decide on the Region where you will work. If the device is already being used as an input, you must work in the current Region. Otherwise, you can choose the Region.
   - If you want to work in a different Region and you can do so, transfer the device now. Then switch the console to that Region. From now on, make sure that you work in this Region.
2. Configure the device. If the device was previously used with a different input or flow, review the current configuration and make any necessary changes. For optimum performance, the device must be correctly configured.

   For more information, see the section called "Configuring a device".
3. Create an Elemental Link input in MediaLive. When you create the input, specify the device as the source. For more information, see the section called "AWS Elemental Link source" and the section called "Elemental Link input".

   As soon as you create the input, the input appears in the **Device details** page, in the **Attachments** tab.
4. When you are ready to use the Elemental Link input in a channel, attach the input to a channel, in the same way as you attach any input. For information, see the section called "The procedure".

   Typically, you attach the input to the channel after the operator at the upstream system has powered on the AWS Elemental Link hardware device, connected it to the internet, and started sending a video stream. You wait to attach the input, in order to avoid charges for an idle input and for a running channel.
5. There are rules for combinations of devices, inputs, and channels. For more information, see *Feature rules and limits*.

Monitor the device

You can use MediaLive to monitor the device.

- You can view thumbnails of the content, if the device is streaming.

- You can look at metrics to monitor the performance of the device.

Using AWS Elemental Link in MediaConnect

You can set up a UHD device as the source for a MediaConnect flow You can't set up an HD device for this usage. For information about setting up in this way, see the section called “Using Link with a MediaConnect flow”.

Implementing automatic input failover

When you set up the inputs for a channel, you can set up two push inputs as an *input failover pair* (or failover pair). Setting up this way provides resiliency for the source, in case of a failure in the upstream system, or a failure between the upstream system and the channel.

You can configure the channel so that MediaLive detects one or more of the following problems in the input:

- Input loss – MediaLive will perform a failover if it doesn't receive content for the specified period.
- Black video (video failure) – MediaLive will perform a failover if content is being received (so input loss doesn't apply), but the content is black for a specified period.
- Audio silence (audio failure) – MediaLive will perform a failover if content is being received (so input loss doesn't apply), but an audio selector is silent for a specified period.

Each input in the input pair provides content to the channel. One of the inputs is the *active* input and one is on *standby*. MediaLive ingests both inputs, in order to always be ready to switch, but it usually discards the standby input immediately. If the active input fails, MediaLive immediately fails over and starts processing from the standby input, instead of discarding it.

> **Note**
>
> Before you decide to implement automatic input failover, you should read about pipeline redundancy, which is another form of channel resiliency. You might decide to implement one or both of these features.

Topics

- Automatic input failover in a single-pipeline channel

- Automatic input failover in a standard channel
- Setting up automatic input failover with CDI inputs
- Setting up automatic input failover with MediaConnect inputs
- Setting up automatic input failover with RTMP and RTP inputs
- Changing the roles of the failover pair
- Starting the channel
- Manually forcing a failover
- Automatic input failover and input switching

Automatic input failover in a single-pipeline channel

You can implement automatic input failover (AIF) in a single-pipeline channel to protect the channel from failure in the upstream system or the network connection that is upstream of MediaLive.

You can implement automatic input failover in push inputs, but not in pull inputs.

Keep in mind that the channel can't have more than two push inputs. This means that you can implement one of these scenarios:

- You can set up one push input in the channel, and you can implement automatic input failover for that input.
- You can set up two push inputs in the channel. In this case, you won't be able to AIF for either of these inputs because you have created the maximum number of push inputs.

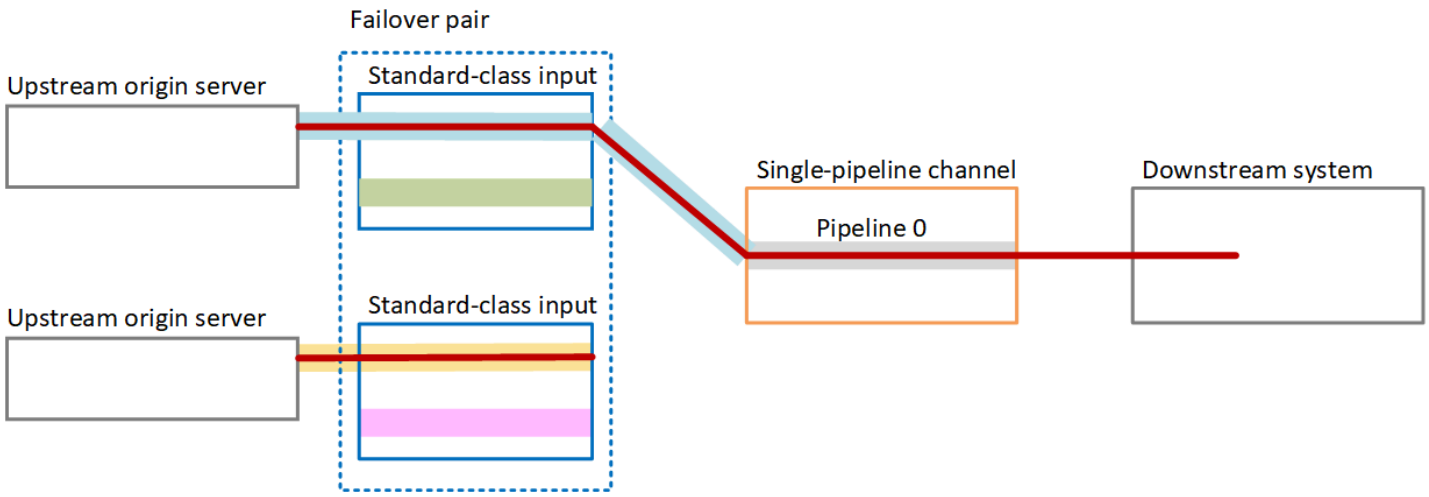
> **Note**
>
> Pay attention to use of the terms *single* and *standard*. The inputs are standard-class. The channel is single-pipeline.

How it works

To implement automatic input failover for the selected push input, you create two standard-class inputs, in the usual way. When you create the channel, you attach these two inputs and then set them up as a failover pair. Both these steps are covered in the setting up secitons later in this topic.

When you start the channel, the channel ingests the content from both inputs. In the diagram, the red lines in the inputs indicate that MediaLive ingests both inputs. But only one input (for example, the blue input in the diagram below) enters the channel pipeline for processing. The other input (the yellow input) is ingested but discarded immediately. The pipeline produces one output for the downstream system, in the usual way.

As this diagram illustrates, there are two instances of the content source.

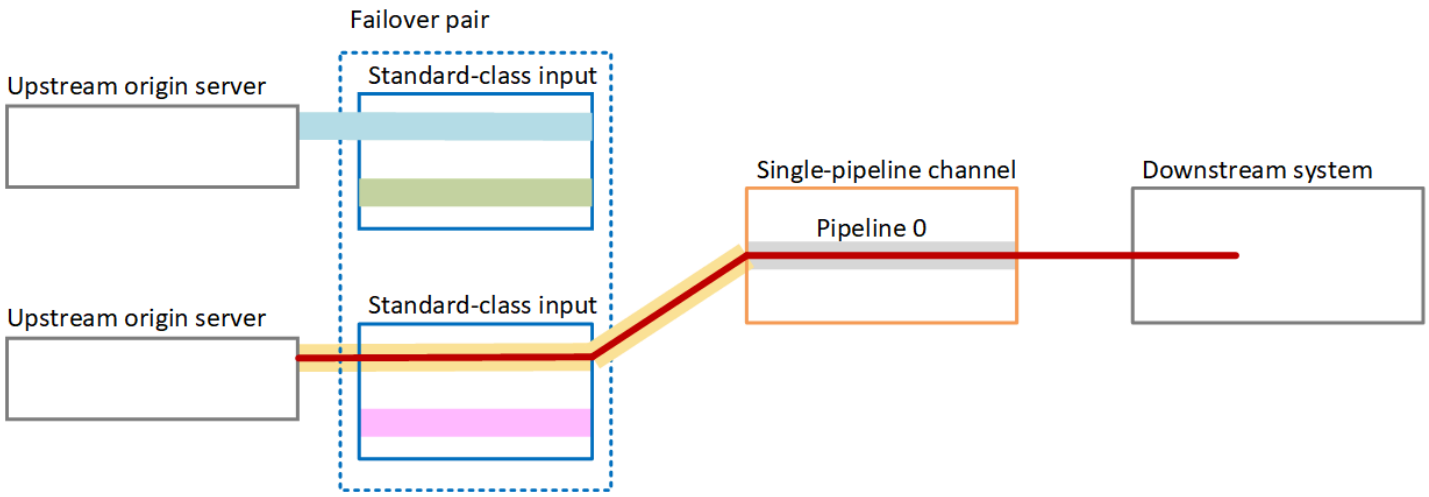


Failure handling

If there is a failure, the behavior is as follows:

- If there is a failure upstream of the first input, then automatic input failover occurs. The channel immediately fails over to the yellow pipeline in the second input, which is already being ingested. The channel fails over and starts processing that input. There is no disruption in the channel pipeline or in the output.
- If there is a failure in the channel pipeline (for example, in pipeline 0), MediaLive stops producing output. Switching the input would not help this failure because the problem is in the pipeline, not in the input.

This diagram illustrates the flow after there is a failure upstream of the first input. MediaLive has failed over to the second input.



Automatic input failover in a standard channel

You can implement automatic input failover in a standard channel to protect the channel from failure in the upstream system or the network connection that is upstream of MediaLive.

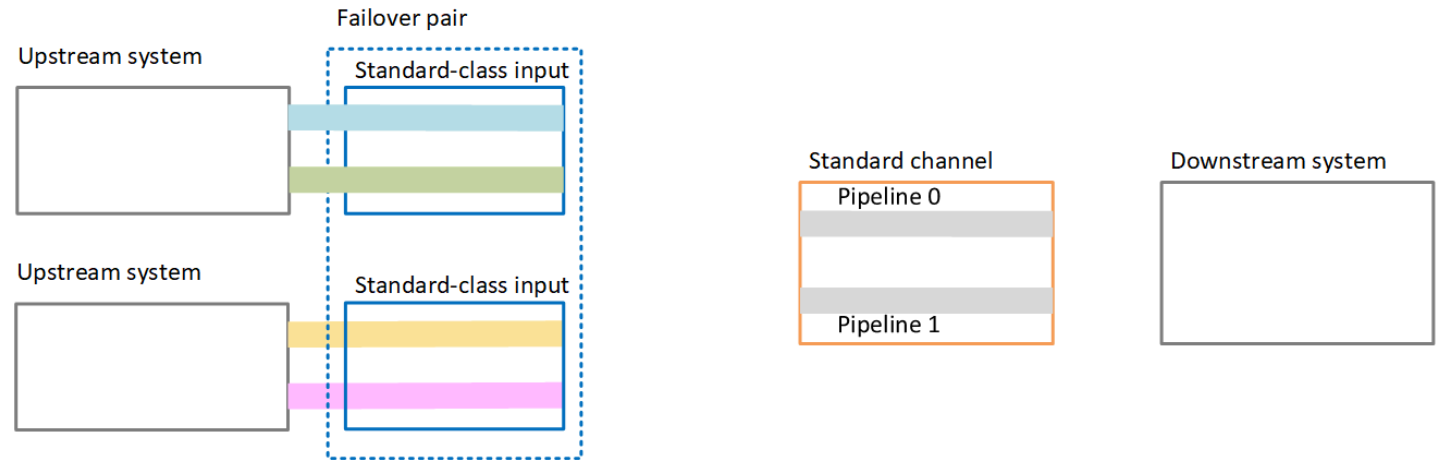
You can implement automatic input failover in push inputs, but not in pull inputs.

Keep in mind that the channel can't have more than two push inputs. This means that you can implement one of these scenarios:

- You can set up two push inputs in the channel, but you won't be able to implement automatic input failover for either of these inputs.
- You can set up one push input in the channel, and you can implement automatic input failover for that one input.

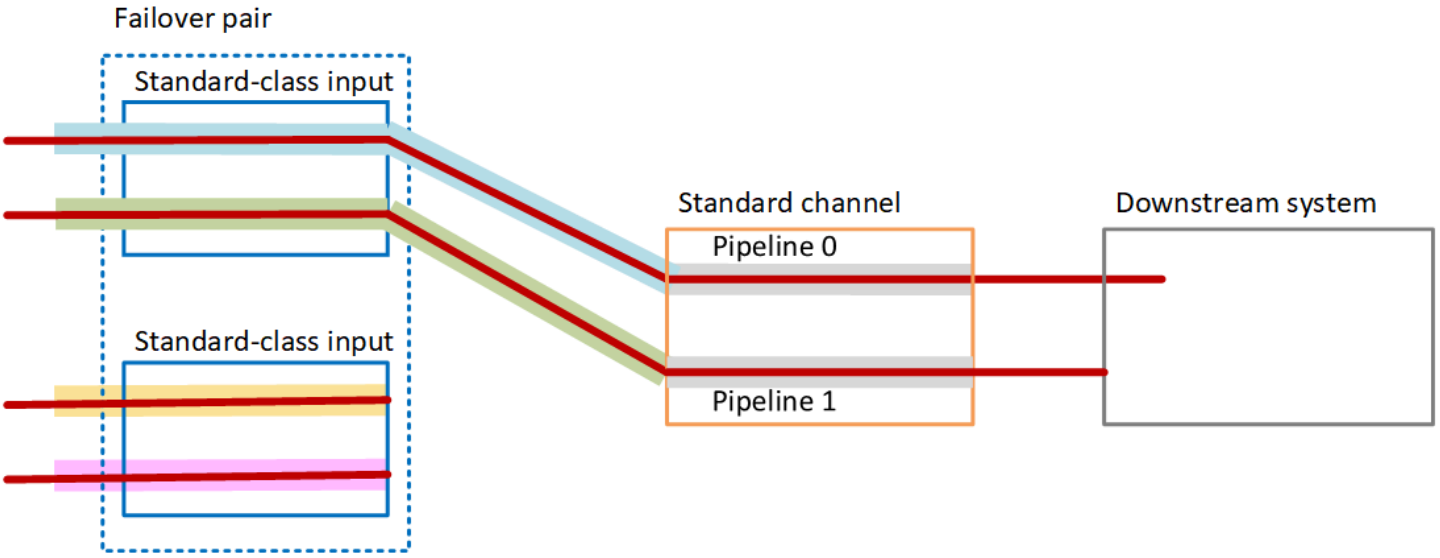
How it works

To implement automatic input failover for the selected push input, you create two standard-class inputs, in the usual way. When you create the channel, you attach these two inputs and then set them up as a failover pair. Both these steps are covered later in the setting up sections later in this topic.



When you start the channel, MediaLive ingests the content from both inputs. So it ingests four sources (as shown by the red lines in the diagram). But only the content from the first input goes to the channel pipeline. The content from the blue pipeline goes to pipeline 0. The content from the green pipeline goes to pipeline 1.

The pipeline produces two outputs for the downstream system, in the usual way. The downstream system chooses to handle one pipeline and to ignore the other pipeline.

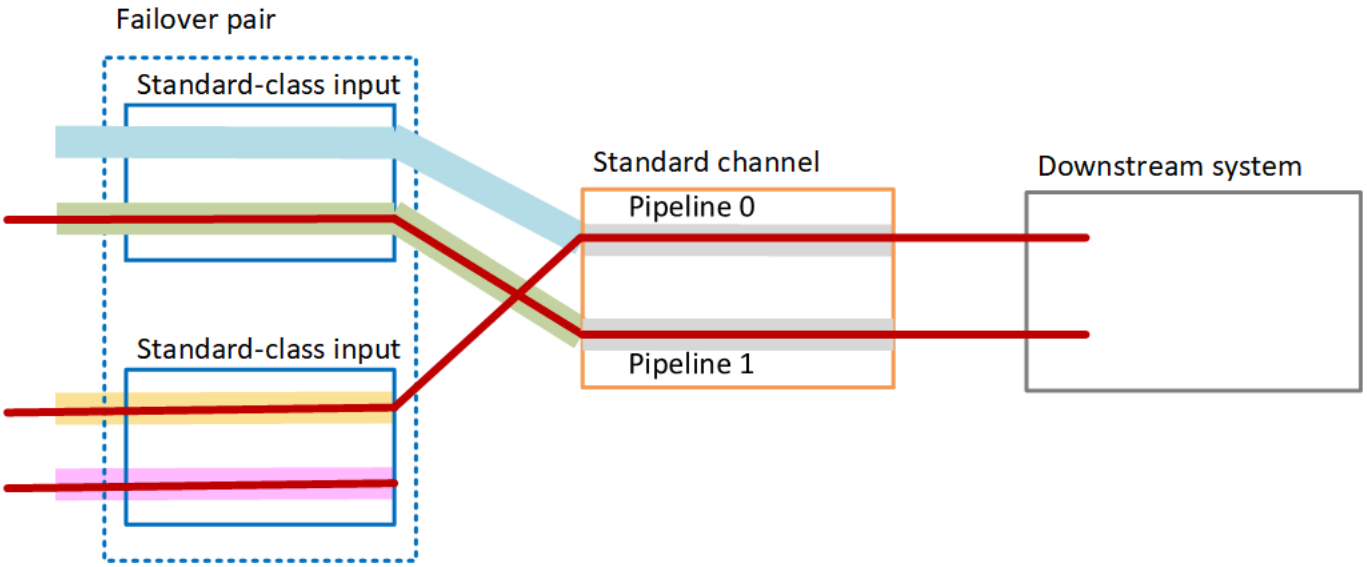


Failure handling

Failure scenario 1

If normal processing is in progress and there is a failure in pipeline 0 in the active input, then the recovery behavior for pipeline redundancy occurs:

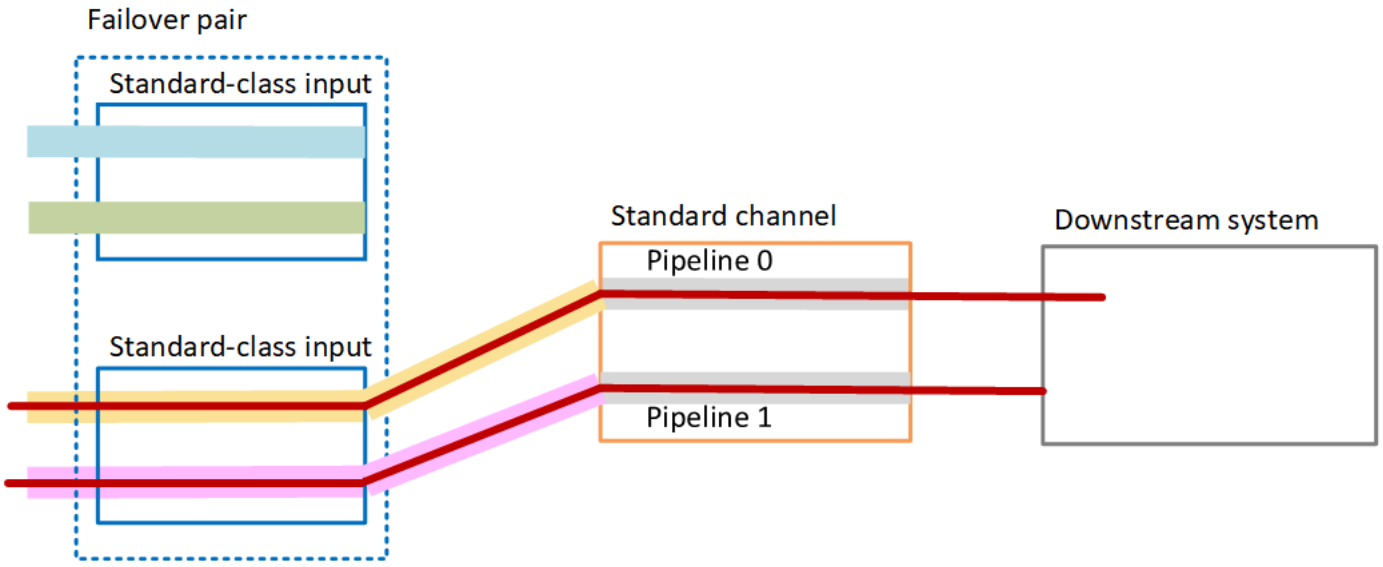
- The channel immediately fails over to pipeline 0 in the second input (which is already being ingested) and starts processing that input. The channel processes the yellow line in pipeline 0, and the green line in pipeline 1. (So there is no change in pipeline 1.) The output is not affected.
- The downstream system continues to handle the output from the pipeline it had chosen before the problem. The downstream system is not affected by the failure in the pipeline 0.



Failure scenario 2

If normal processing is in progress and there is a failure upstream of the first input, then automatic input failover occurs:

- The channel immediately fails over to the second input (which is already being ingested) and starts processing that input. The yellow line is processed in pipeline 0, the pink line in pipeline 1. The output is not affected.
- The downstream system continues to handle the output from the pipeline it had chosen before the problem. The downstream system is not affected by the failure in the first input.



Setting up automatic input failover with CDI inputs

To use CDI inputs with automatic input failover, you must make sure that the upstream system provides sources in the correct way, and you must set up the inputs and the channels in a specific way.

> **Note**
>
> The information in this section assumes that you are familiar with the general steps for creating an input and creating a channel.

To plan the inputs for the input failover pair

1. Arrange with your upstream system for them to provide you with the appropriate number of sources for the content:
   - If you are setting up automatic input failover in a single-pipeline channel, you need two sources—one for each input.
   - If you are setting up automatic input failover in a standard channel, you need four sources—two for each input.
2. Make sure that the upstream system sets up the paths correctly. The first input must have a different network path to MediaLive, compared to the second input. MediaLive can't enforce this rule, but the point of automatic input failover is that the sources arrive via different paths.

If they don't, then when the route fails, both inputs will fail, and you will not have achieved redundancy.

3. Make sure that the input type for the sources is CDI.
4. Make sure that all the sources contain exactly identical video, audio, captions, and metadata.

To create the inputs for the input failover pair

- Create a set of two partner CDI inputs. See the section called “CDI input – Partner CDI input”.

  Don't follow the usual procedure of creating two independent CDI inputs. You won't be able to set up these two inputs as a failover pair.

To attach the inputs to the channel

1. Decide which partner CDI input you want to set up as the primary input.
2. In the **Input attachments** section of the **Create channel** page, follow the usual procedure to attach the primary input. Ignore the **Automatic input failover settings** for now.

   Remember to set up the **General settings**, particularly the selectors.
3. Follow the same procedure as the previous step to attach the partner input.
4. In the **Input attachments** section, in the list of input attachments, choose the first input that you attached.

   You must choose the first input that you attached. If you choose the other partner input, you won't be able to enable automatic input failover.
5. In the **Automatic input failover settings** section, choose **Enable automatic input failover settings**. As soon as you enable this field, this input is labeled as **Primary** in the list of input attachments.
6. For **Secondary input**, choose the partner input. The partner input is the only input in the list. If no inputs are listed, you have forgotten to create the partner input. Create it now.
7. For **Input preference**, choose the desired option. This field controls the behavior when MediaLive has switched over to the secondary input and then the primary input becomes healthy again.

   - **EQUAL_INPUT_PREFERENCE** – MediaLive remains on the secondary input. The primary input continues to be processed, but it is not active.

- **PRIMARY_INPUT_PREFERENCE** – MediaLive switches back to the primary input. The primary input becomes the active input.

8. For **Failover conditions**, enable the conditions that you want MediaLive to use to identify input loss. The fields include help that describes how the conditions work.

Setting up automatic input failover with MediaConnect inputs

To use MediaConnect inputs with automatic input failover, you must set up both the inputs and the channels in a specific way.

> **Note**
>
> The information in this section assumes that you are familiar with the general steps for creating a MediaConnect input and creating a channel.

To plan the inputs for the input failover pair

1. Identify the flows that you need to create on MediaConnect:
   - If you are setting up automatic input failover in a single-input channel, you need two flows—one for each input.
   - If you are setting up automatic input failover in a standard channel, you need four flows—two for each input.
2. Make sure that all the flows contain exactly identical video, audio, captions, and metadata.

To create the flows in MediaConnect in a standard channel

You must create four flows, two for the primary input, and two for the secondary input.

- Follow the procedure in the section called “Step 2: Create input”, with the following notes:

  Make sure that you set up the flows in the correct Availability Zones. Assume that the two flows for the primary input are A and B, and that the two flows for the secondary input are C and D.

  - Flow A must be in Availability Zone X.
  - Flow B must be in Availability Zone Y.

- Flow C must be in Availability Zone X.
- Flow D must be in Availability Zone Y.

At channel startup, MediaLive sets up the flows as follows:

- Flow A connects to pipeline 0.
- Flow C connects to pipeline 0.

- Flow B connects to pipeline 1.
- Flow D connects to pipeline 1.

As a result of these connections, the active input on pipeline 0 is initially from Availability Zone X. The active input on pipeline 1 is initially from Availability Zone Y. If one Availability Zone fails, only one pipeline is affected. For more information on failure scenarios, see the section called “Failover and failback scenarios”.

To create the flows in MediaConnect in a single-pipeline channel

You must create two flows, one for each input.

- Follow the procedure in the section called “Step 2: Create input”, with the following note:

  Make sure that you set up the flows in the same Availability Zones. The two inputs provide two paths to the single pipeline in the channel. If one of the flows fails to send content, that input fails and MediaLive switches to the other input.

To create the inputs for the input failover pair

1. Follow the procedure in the section called “Step 2: Create input” to create one input of the appropriate type.
   - In a standard channel, set up the input with two sources. Attach flows A and B to this input.
   - In a single-pipeline channel, set up the input with one flow.
   - Give the input a name such as **`primary input`**.
2. Create a second input in the same way.

- In a standard channel, set up the input with two sources. Attach flows C and D to this input.
- In a single-pipeline channel, set up the input with one flow.
- Give the input a name such as **`secondary input`**.

To attach the inputs to the channel

1. In the **Input attachments** section of the **Create channel** page, follow the usual procedure to attach the primary input. Ignore the **Automatic input failover settings** for now.
2. Follow the same procedure to attach the secondary input.
3. In the **Input attachments** section, in the list of input attachments, choose the first input that you attached.
4. In the **Automatic input failover settings** section, choose **Enable automatic input failover settings**. As soon as you enable this field, this input is labeled as **Primary** in the list of input attachments.
5. For **Secondary input**, choose the secondary input. (When you do this, this input is labeled as **Secondary** in the list of attachments.)
6. For **Input preference**, choose the desired option. This field controls the behavior when MediaLive has switched over to the secondary input and then the primary input becomes healthy again.

   - **EQUAL_INPUT_PREFERENCE** – MediaLive remains on the secondary input. The primary input continues to be processed, but it is not active.
   - **PRIMARY_INPUT_PREFERENCE** – MediaLive switches back to the primary input. The primary input becomes the active input.
7. For **Failover conditions**, enable the conditions that you want MediaLive to use to identify input loss. The fields include help that describes how the conditions work.

   Note

   If you enable the input loss failover condition, find out if the MediaConnect flow implements source redundancy with failover mode. With this mode, if there is a source failure, MediaConnect waits 500 ms for the source to recover before it fails over. Therefore, you must configure MediaLive to wait longer than 500ms, to ensure that MediaLive doesn't fail over just as MediaConnect is about to recover.

In the **Enable input loss settings** option, adjust the threshold. Set the threshold to a value higher than 500ms. You might need to try different values to find the ideal threshold for your network.

Setting up automatic input failover with RTMP and RTP inputs

To use RTMP push inputs and RTP inputs with automatic input failover, you must make sure that the upstream system provides sources in the correct way, and you must set up the inputs and the channels in a specific way.

Note

The information in this section assumes that you are familiar with the general steps for creating an input and creating a channel.

To plan the inputs for the input failover pair

1. Arrange with your upstream system for them to provide you with the appropriate number of sources for the content:
   - If you are setting up automatic input failover in a single-input channel, you need two sources—one for each input.
   - If you are setting up automatic input failover in a standard channel, you need four sources—two for each input.
2. Make sure that the upstream system sets up the paths correctly. The first input must have a different network path to MediaLive, compared to the second input. MediaLive can't enforce this rule, but the point of automatic input failover is that the sources arrive via different paths. If they don't, then when the route fails, both inputs will fail, and you will not have achieved resiliency.
3. Make sure that the input type for the sources is the same. For example, two RTMP inputs.
4. Make sure that all the sources contain exactly identical video, audio, captions, and metadata.

To create the inputs for the input failover pair

1. Follow the procedure in the section called "Inputs" to create one input of the appropriate type. For example, one RTMP input.

   - In a standard channel, set up the input with two sources.
   - In a single-pipeline channel, set up the input with one source.
   - Give the input a name such as **`primary input`**.

2. Create a second input of the same type. Create the input in the same way as in Step 1.

   Give the input a name such as **`secondary input`**.

To attach the inputs to the channel

1. In the **Input attachments** section of the **Create channel** page, follow the usual procedure to attach the primary input. Ignore the **Automatic input failover settings** for now.
2. Follow the same procedure to attach the secondary input.
3. In the **Input attachments** section, in the list of input attachments, choose the first input you attached.
4. In the **Automatic input failover settings** section, choose **Enable automatic input failover settings**. As soon as you enable this field, this input is labeled as **Primary** in the list of input attachments.
5. For **Secondary input**, choose the secondary input. (When you do this, this input is labeled as **Secondary** in the list of attachments.)
6. For **Input preference**, choose the desired option. This field controls the behavior when MediaLive has switched over to the secondary input and then the primary input becomes healthy again.

   - **EQUAL_INPUT_PREFERENCE** – MediaLive remains on the secondary input. The primary input continues to be processed, but it is not active.
   - **PRIMARY_INPUT_PREFERENCE** – MediaLive switches back to the primary input. The primary input becomes the active input.

7. For **Failover conditions**, enable the conditions that you want MediaLive to use to identify input loss. The fields include help that describes how the conditions work.

Changing the roles of the failover pair

You can reverse the roles of the two inputs, so that the primary input becomes the secondary input.

To reverse the roles of the inputs

1. From the list of input attachments, choose the first input that you attached.
2. In the **Automatic input failover settings** section, choose **Disable automatic input failover settings**.
3. Choose the second input and choose **Enable automatic input failover settings** for that input. The second input is now the primary input.

Starting the channel

Start the channel in the usual way. MediaLive follows this behavior when you start the channel:

- If the input attachment list contains only the input failover pair, MediaLive starts with the primary input, which always appears first in the attachments.
- If you have set up the channel to always use the schedule, even with the first input, then MediaLive starts with the first input in the schedule. This input can be any input.
- If you have not set up the channel to control startup behavior (not recommended), MediaLive starts with the first input in the input attachment list.

Failover and failback scenarios

Failover follows this rule:

- If the active input is unhealthy for 3 seconds, MediaLive switches to the other input.

You can also manually switch to the other input, if the **Input preference** setting is **EQUAL_INPUT_PREFERENCE**. Switching over manually is useful, for example, if you believe that the active input is unstable. See the section called “Manually forcing a failover”.

Failback follows this rule:

- When the unhealthy input is healthy again for more than 30 seconds, it is marked as healthy.

When the input becomes healthy, MediaLive might automatically switch to the healthy input:

- If the currently active input is the secondary input, MediaLive either stays on the current input (if the **Input preference** setting is **EQUAL_INPUT_PREFERENCE**) or switches to the primary input (if the **Input preference** setting is **PRIMARY_INPUT_PREFERENCE**).
- If the active input is the primary input, it always stays on the input.

Manually forcing a failover

You can set up automatic input failover for manual failover.

Keep in mind that the content in the failover pair is identical. Therefore, you only switch between them for specific reasons. For example:

- You might think that the active input is degrading, but MediaLive hasn't yet made the decision to fail over to the other input.
- You might want to perform maintenance on the network for the input that is currently active.

To switch between the two inputs in the input pair

1. If you think you might want to manually switch inputs, then when you set up the failover pair, set the **Input preference** to **EQUAL_INPUT_PREFERENCE**. See the section called “Setting up: other inputs” or the section called “Setting up: MediaConnect inputs”.
2. To manually switch, create an input switch action in the schedule in the usual way.

   Set up the input to switch to the other input, and set the **Start Type** to **Immediate**.

Automatic input failover and input switching

When you implement automatic input failover, you can still implement input switching.

> **Note**
>
> The information in this section assumes that you are familiar with the general steps for creating input switches, as described in the section called “Creating actions”.

With automatic input failover, your deployment contains an input failover pair that uses up your quota of push inputs for the channel. You can't attach more push inputs to the channel. But you can attach more pull inputs, and can therefore set up a multiple-input channel suitable for input switching using the schedule. You can perform the following switches:

- From a pull input to another pull input.
- From a pull input to either input in the failover pair.
- From the primary input or secondary input to a pull input.

Including captions in an AWS Elemental MediaLive channel

You can set up the AWS Elemental MediaLive channel to extract captions when it ingests the source, and to include those captions in the output in either the same or a different format. You can include several captions in the output. For example, you can include captions for several languages. You can take a source captions asset and convert it to one format in one output and to another format in a different output.

You perform the setup for captions in your AWS Elemental MediaLive channel.

By default, AWS Elemental MediaLive does not ingest any captions (not even captionsa that are embedded in the video). You must explicitly identify the captions to ingest and the captions to output.

> **Note**
>
> The information in this captions section assumes that you are familiar with the general steps for creating a channel, as described in *Setup: Creating a channel* . It also assumes that you have started creating a channel, including associating an input with the channel.

Topics

- Step 4: Create captions encodes in AWS Elemental MediaLive
- Examples of handling captions in AWS Elemental MediaLive

Captions features supported in an AWS Elemental MediaLive channel

This section provides information on the different features of captions that AWS Elemental MediaLive supports.

Topics

Captions formats supported in an AWS Elemental MediaLive channel

AWS Elemental MediaLive supports specific formats in inputs and specific formats in outputs. See the section called "Supported formats" for a table that lists the supported captions formats, with a reference to the standard that defines that format. The table specifies whether the format is supported as input or output or both.

Formats supported in different types of AWS Elemental MediaLive outputs

There are several factors that control your ability to include captions of a specific format in your outputs:

- **The type of input container** – A given input container can contain captions in some formats and not in others.
- **The format of the input captions** – A given format of captions can be converted to some formats and not to others.
- **The type of output containers** – A given output container supports some captions formats and not others.

For example, assume that your input container is an MP4 container and your output is HLS, and that you want to include WebVTT captions in the HLS output. You can implement this use case

only if the MP4 container holds 608 embedded captions. You can't implement it if, for example, the MP4 container holds Ancillary captions.

For more information about all the supported combinations of input container, input format, and output container, see the section called “Captions: Supported formats”.

Constraints for using OCR conversion with captions in AWS Elemental MediaLive

MediaLive uses OCR (optical character recognition) technology for the following scenarios:

- The input captions are DVB-Sub or SCTE-27
- The output captions are WebVTT format

Constraint in supported languages

OCR conversion uses language libraries. Language libraries are a critical component of conversion. They speed up conversion because the tool can check character strings against a dictionary, instead of recognizing words letter by letter. You must specify the language of a captions source so that MediaLive can choose the correct library. If you choose a language that doesn’t match the language of the captions, conversion accuracy will be poor.

MediaLive currently includes libraries for six languages, which means that MediaLive can perform an OCR conversion only with the following source languages:

- Dutch
- English
- French
- German

- Portuguese
- Spanish

Constraint in number of languages in one input

OCR conversion uses more processing resources than other captions conversions. Therefore, in each input, you can create a maximum of three captions selectors that will use OCR conversion.

These rules apply:

- A selector uses OCR conversion if the specified format is DVB-Sub or SCTE-27, and at least one output encode that uses the selector is a WebVTT encode.

- A DVB-Sub or SCTE-27 selector doesn't use OCR conversion (and doesn't count towards the limit) if, for example, the selector is used only in SMPTE-TT encodes.
- If the selector is used in more than one WebVTT encode (for example, in two output groups), the selector counts only once towards the limit.

Support for multiple captions languages in AWS Elemental MediaLive

Some sources typically include multiple languages. You can include multiple languages in the output as follows.

Source captions	Output captions	Result
ARIB	ARIB (which is the only option for ARIB sources)	This combination is performed as a passthrough. Therefore, all the languages that are in the input are included in the output. You can't remove any languages.
Embedded	Embedded	This combination is performed as a passthrough. Therefore, all the languages that are in the input are included in the output. You can't remove any of the languages.
Embedded	Another format	You can specify which languages to extract and include in an output.
DVB-Sub	Another format	If you have DVB-Sub source and want a different format in the output, you can specify

Source captions	Output captions	Result
		the language to extract from the input and the language to include in an output.
DVB-Sub	DVB-Sub	This combination is performed as a passthrough. Therefore, all the languages that are in the input are included in the output. You can't remove any languages.
Teletext	Teletext	This combination is performed as a passthrough. Therefore, all the languages that are in the input are included in the output. You can't remove any languages. In fact, the entire Teletext content is included in the output; you can't remove out any of the pages.
Teletext	Another format	You can specify the language to extract from the input and the language to include in an output.
Any other combination		For all other sources, you always specify the language to extract from the input and the language to include in an output, regardless of the source format and output format.

Support for font styles in output captions in AWS Elemental MediaLive

Depending on the scenario, there are three possibilities for the font style for output captions:

- You can specify the style that you want for fonts, including color, outline, and background color.
- The font styles in the input are passed through.
- The font styles are controlled by the downstream player.

The procedures later in this chapter describe how to set up font styles. You might set up the styling of the output captions on the input side, on the output side, or on both sides.

Source captions	Output captions	Options for font style
ARIB	ARIB	None. The font styles in the input are automatically passed through in the output.
Any supported captions format	Burn-in	You can specify font styles in the output. If you don't specify styles, the AWS Elemental MediaLive defaults are used.
DVB-Sub	DVB-Sub	None. The font styles in the input are automatically passed through in the output.
Any supported captions format	DVB-Sub	You can specify font styles in the output. If you don't specify styles, the AWS Elemental MediaLive defaults are used.
An Embedded Combination (Embedded, Embedded +SCTE-20, SCTE-20+Embedded)	EBU-TT-D	You can specify some of the style information and take some of the information from the input captions. Or you can

Source captions	Output captions	Options for font style
		set up the captions with no style data.
Teletext	EBU-TT-D	You can specify some of the style information and take some of the information from the input captions. Or you can set up the captions with no style data.
Teletext	Teletext	None. The font styles in the input are automatically passed through in the output.
An Embedded Combination (Embedded, Embedded +SCTE-20, SCTE-20+Embedded)	TTML	You can set up to copy the font information from the source to the output. Or you can let the downstream player determine the font style.
Teletext	TTML	You can set up to copy the font information from the source to the output. Or you can let the downstream player determine the font style.
An Embedded Combination (Embedded, Embedded +SCTE-20, SCTE-20+Embedded)	WebVTT	You can set up to pass through color and position style information from the source to the output. Or you can set up the captions with no style data.

Source captions	Output captions	Options for font style
Teletext	WebVTT	You can set up to pass through color and position style information from the source to the output. Or you can set up the captions with no style data.
Any Other	Any Other	No control: the font style is always determined by the downstream player.

Typical scenarios for handling captions in AWS Elemental MediaLive

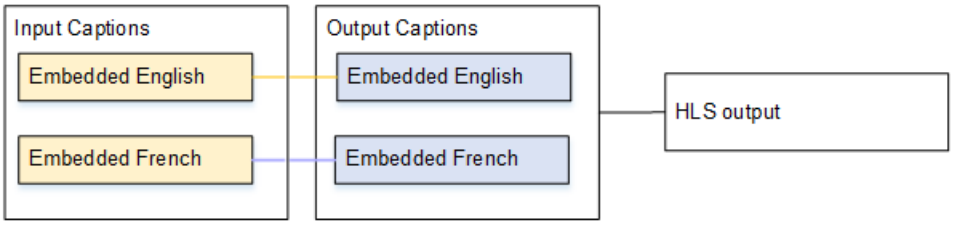
Following are some sample use cases. The use cases are ordered from less to more complicated. They are intended to illustrate many of the capabilities of MediaLive.

Topics

Use case A: One input format to one output and not converted

The input is set up with one format of captions and two or more languages. Assume that you want to maintain the format in the output, and that you want to produce only one type of output and to include all the languages in that output.

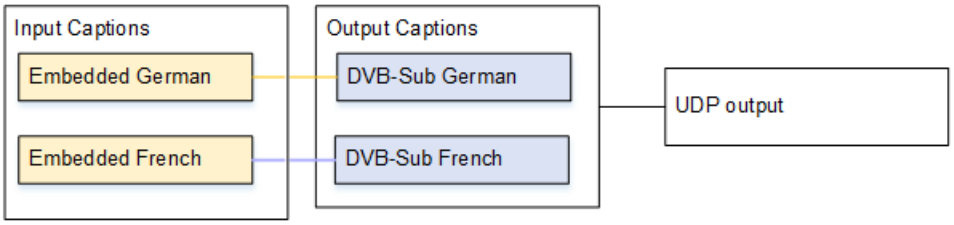
For example, the input has embedded captions in English and French. You want to produce HLS output that includes embedded captions in both English and French.



Use case B: One input format converted to one different format in one output

The input is set up with one format of captions and two or more languages. You want to convert the captions to a different format in the output. You want to produce only one type of output and include all the languages in that output.

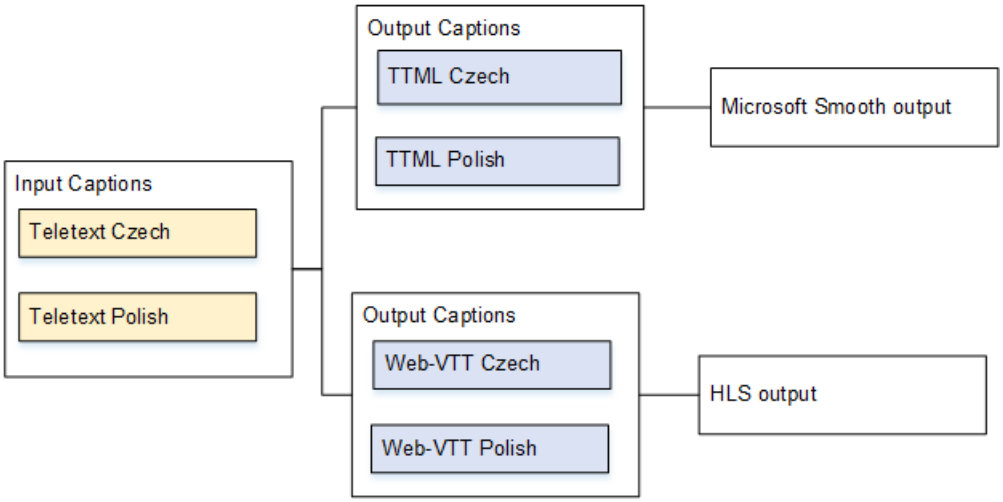
For example, the input has embedded captions in German and French. You want to convert the captions to DVB-Sub and include these captions in both languages in a UDP output.



Use case C: One input format converted to different formats, one format for each output

The input is set up with one format of captions and two or more languages. Assume that you want to produce several different types of output, and that in each output you want to convert the captions to a different format but include all the languages.

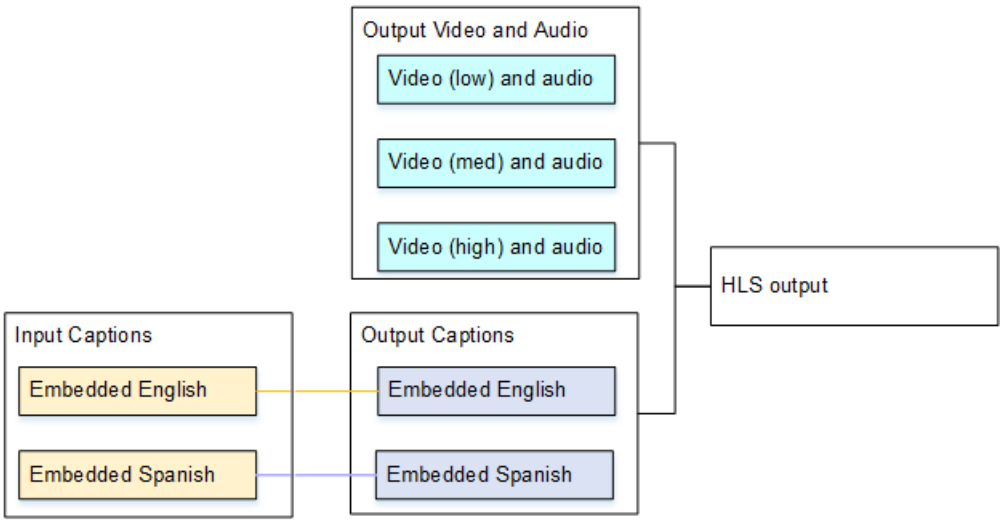
For example, the input has Teletext captions in Czech and Polish. You want to produce a Microsoft Smooth output and an HLS output. In the Microsoft Smooth output, you want to convert both captions to TTML. In the HLS output, you want to convert both captions to WebVTT.



Use case D: One captions output shared by multiple video encodes

This use case deals with captions in an ABR workflow.

For example, assume that there are three video/audio media combinations: one for low-resolution video, one for medium, and one for high. Assume that there is one output captions asset (English and Spanish embedded) that you want to associate with all three video/audio media combinations.



Step 1: Create captions selectors in the AWS Elemental MediaLive inputs

You must identify the captions that you want to use and assign each to a captions selector. If you don't create any captions selectors, you can't include captions in the output. All the captions will be removed from the media.

Then you must extract the captions that you want by adding a captions selector in the channel. Each extracted captions asset is contained in one captions selector. For example, one selector contains the Teletext captions in Czech.

Topics

Identify the captions that you want

1. Identify which captions are in the input (the provider of the input should provide you with this information). Identify the captions formats and, for each format, the languages.
2. Identify which of those formats and languages that you want to use.
3. If you are converting DVB-Sub or SCTE-27 captions to WebVTT, there are limits on the number of languages that MediaLive can ingest. For more information, see the section called "Constraints with OCR conversion".
4. Determine how many captions selectors to create in the input in the channel. For guidance, see the table that appears after this procedure.

You end up with a list of captions selectors to create. For example:

- Captions Selector 1: Teletext captions in Czech
- Captions Selector 2: Teletext captions in Polish

Source captions	Output captions	Result
ARIB	ARIB	Create a single captions selector. All languages are passed through; there is no other option.

Source captions	Output captions	Result
Embedded	Embedded	Create a single captions selector. All languages are passed through; there is no other option. For details, see the section called “Information for embedded ”.
Embedded	Another format	Specify the language to extract from the input and the language to include in an output. The specified language is extracted from the embedded captions and converted to the new format.
DVB-Sub	WebVTT	Create one caption selector for each language, to a maximum of three caption selectors in the input. For more information about this limit, see the section called “Constraints with OCR conversion”.
DVB-Sub	SMPTE-TT	Create one caption selector for each language. The specified language is extracted from the DVB-Sub captions and converted to the new format.
DVB-Sub	DVB-Sub	Create a single captions selector. All languages are passed through.

Source captions	Output captions	Result
SCTE-27	WebVTT	Create one caption selector for each language, to a maximum of three caption selectors in the input. For more information about this limit, see the section called "Constraints with OCR conversion".
Teletext	Teletext	Create a single captions selector. All languages are passed through. All the pages in the teletext are passed through. For details, see the section called "Information for Teletext".
Teletext	Another format	If you have Teletext source and want a different format in the output, create one captions selector for each language and format combination.
Any other combination		Create one captions selector for each language and format combination.

Create captions selectors

1. In the channel that you are creating, in the navigation pane, in **Input attachments**, choose the input.
2. Got to **General input settings** and choose **Add captions selectors**.

3. In **Captions selector name**, enter a name that describes the captions in the source.. For example, **`Teletext Czech`**.
4. In **Selector settings**, choose the format of the source captions.
5. For most formats, more fields appear. For details about a field, choose the **Info** link next to the field. In addition, see DVB-Sub or SCTE-27, Embedded, or Teletext.
6. Create more captions selectors, as required.

Information for DVB-Sub or SCTE-27

1. You must specify the location of the captions.

   Complete the **PID** or **Language** code fields in one of the ways described in the following table. Each row in the table describes a valid way to complete these two fields.

PID	Language Code	Result
Specified	Blank	Extracts captions from the specified PID.
Blank	Specified	Extracts the captions from the first PID that MediaLive encounters that matches the specified language. This might or might not be the PID with the lowest number.
Specified	Specified	Extracts the captions from the specified PID. MediaLive ignores the language code, therefore we recommend you leave it blank.
Blank	Blank	Valid only if the source is DVB-Sub and the output is DVB-Sub. With this combination of PID and Language, all input DVB-

PID	Language Code	Result
		Sub PIDs are included in the output. Not valid for SCTE-27.

2. If you plan to convert the captions to WebVTT, you must also specify the language of the captions.

   Complete the **OCR language** field to specify the language of the captions specified by this selector.

   MediaLive ignores any value in this field if you aren't converting the captions to WebVTT.

Information for embedded

Read this section if the input captions are any of the following: embedded (EIA-608 or CEA-708), embedded+SCTE-20, SCTE-20+embedded, or SCTE-20.

How many captions selectors?

- **Embedded passthrough** – Create only one captions selector. With this scenario, all languages are automatically extracted and are automatically included in the output.
- **Embedded in, other out** – Create one captions selector for each language that you want to include in the output, to a maximum of four selectors.
- **A combination of Embedded passthrough and Embedded conversion** – If you are setting up for embedded passthrough in some outputs and embedded-to-other in other outputs, create one captions selector for each language that you want to include in the output, to a maximum of four selectors. Don't worry about a selector for the embedded passthrough output. MediaLive extracts all the languages for that output, even though there is not a selector to explicitly specify this action.

Captions selector fields

- **Selector settings**:
  - Choose embedded if the source captions are embedded (EIA-608 or CEA-708), embedded +SCTE-20, or SCTE-20+embedded.

- Choose SCTE-20 if the source captions are SCTE-20 alone.

- **EIA-608 track number** – This field specifies the language to extract. Complete as follows:
  - If you are setting up for embedded passthrough only (you are creating only one captions selector for the input embedded captions), this field is ignored, so keep the default.
  - If you are converting embedded to another format (you are creating several captions selectors, one for each language), specify the number of the CC instance (from the input) that holds the language that you want.

- **Convert 608 to 708**: The embedded source captions can be EIA-608 captions, CEA-708 captions, or both EIA-608 and CEA-708. You can specify how you want these captions to be handled when AWS Elemental MediaLive is ingesting content. The following table describes the behavior for various scenarios.

EIA-608 in source	CEA-708 in source	Convert field	Result
Yes	No	**Upconvert**	CEA-708 data is created based on the EIA-608 data. EIA-608 data is added as 608-compatibility bits in the CEA-708 data.
Yes	No	**Disabled**	Original EIA-608 is preserved.
No	Yes	**Upconvert**	Original CEA-708 is preserved.
No	Yes	**Disabled**	Original CEA-708 is preserved.

EIA-608 in source	CEA-708 in source	Convert field	Result
Yes	Yes	**Upconvert**	CEA-708 data is discarded. New CEA-708 data is created based on the EIA-608 data. EIA-608 data is added as 608-compatibility bits in the CEA-708 data. The new CEA-708 data will not include any CEA-708 formatting features. Not recommended.
Yes	Yes	**Disabled**	Original EIA-608 is preserved and original CEA-708 is preserved.

- **SCTE-20 detection** – If the source captions combine embedded (EIA-608 or CEA-708) and SCTE-20, you might want to set this field to **Auto**. AWS Elemental MediaLive gives preference to the 608/708 embedded captions but switches to use the SCTE-20 captions when necessary. If you set this field to **Off**, AWS Elemental MediaLive never uses the SCTE-20 captions.

Information for Teletext

Teletext is a form of data that can contain several types of information, not just captions. Teletext can be handled in one of the following ways:

- If you want to include the entire Teletext input, you must set up for Teletext passthrough. The entire Teletext can never be converted to another format.
- Individual captions pages (the captions in a specific language) can be extracted and converted to another captions format.

- Individual captions pages (the captions in a specific language) *cannot* be extracted and kept in Teletext. If you want to extract individual captions pages, you must convert them to another format.

How many captions selectors?

- If you are setting up for Teletext passthrough captions, create only one captions selector, even if you want to include multiple languages in the output. With this scenario, all languages are automatically extracted and included in the output.
- If you are setting up for Teletext-to-other, create one captions selector for each language that you want to include in the output. For example, one selector to extract English Teletext, and one selector to extract Swedish Teletext.
- If you are setting up for Teletext passthrough in some outputs and Teletext-to-other in other outputs, create one captions selector for each language that you want to include in the output. Don't worry about a selector for the passthrough output. MediaLive passes through all the data, even though there isn't a selector to explicitly specify this action.

Captions selector fields

- **Selector settings** – Choose **Teletext**.
- **Page number** – This field specifies the page of the desired language. Complete as follows:
  - If you are setting up for Teletext passthrough captions (you are creating only one captions selector for the input captions), keep the field blank. The value is ignored.
  - If you are converting Teletext to another format (you are creating several captions selectors, one for each language), complete the **Language code** field to specify the page for the language that you want. If you leave this field blank, you get a validation error when you save the channel.

Including a positioning rectangle

If you plan to convert the source captions to EBU-TT-D, you can optionally define a rectangle that positions the captions on the video frame in the output. If you choose to use this feature, it will apply as follows:

- It will apply to all your EBU-TT-D outputs that use this captions selector.

- It won't apply to any other formats of output captions that use this caption selector. The positioning information is simply omitted from these other captions formats.

You define the rectangle relative to the underlying video frame. For example, you specify the position of the left edge of the rectangle as a percentage of the entire width of the video frame. A value of 10 means "calculate a value X that is 10% of the frame width. Then find the left edge of the video frame and move X pixels into the frame and draw the left edge of the rectangle".

Specifying a percentage, rather than a fixed number, means that the rectangle works for different video renditions (different resolutions) in the same output.

To define a positioning rectangle, follow this procedure.

1. In the **Output rectangle** field, choose **Caption rectangle**.
2. Complete the fields for the four sides of the rectangle – **Left offset**, **Width**, **Top offset**, and **Height**.

Step 2: Plan captions for the AWS Elemental MediaLive outputs

If you followed the instructions in the section called “Step 1: Set up inputs”, you should have a list of the captions formats and languages that are available for inclusion in the outputs.

You must now plan the captions information for the outputs.

To plan the captions for the output

- Identify the types of output media that you plan to create in the channel, for example, Microsoft Smooth and HLS.
- Identify the combinations of video and audio that you plan to create for each output media.
- For each output media, identify which input captions will be converted to which output formats. For example, you will convert Teletext captions to TTML for the Microsoft Smooth output media, and those same Teletext captions to WebVTT for the HLS output media.

  The output formats that are possible depend on the input formats and the type of output media. To determine which output captions are possible given the input format, see the section called “Captions: Supported formats”.
- Identify the languages for each output format:

- In general, count each language separately.
- Exception: For embedded passthrough, count all languages as one.
- Exception: For Teletext passthrough, count all languages as one.

The result

You end up with a list of outputs, and the captions formats and languages for each output. For example:

- Microsoft Smooth output with TTML captions in Czech
- Microsoft Smooth output with TTML captions in Polish
- HLS output with WebVTT captions in Czech
- HLS output with WebVTT captions in Polish

Outputting multiple formats

You can include captions from two or more different formats in an output. For example, you can include both embedded captions and WebVTT captions in an HLS output, to give the downstream system more choices about which captions to use. The only rules for multiple formats are the following:

- The output container must support all the formats. See the section called “Captions: Supported formats”.
- The font styles in all the captions that are associated with an output must match. This means that the end result must be identical, not that you must use the same option to get that result. For example, all captions that are associated with the output must be white for the first language and blue for the second language.

Managing this style matching can be a little tricky. For information about the font style options, see Support for font styles in output captions in AWS Elemental MediaLive.

Step 3: Match formats to categories

There are different procedures to follow to create captions encodes in the output. The correct procedure depends on the "category" that the output captions belong to. There are five categories of captions, described in the section called “Captions categories”.

On the list of outputs that you have created, make a note of the category that each captions option belongs to.

Step 4: Create captions encodes in AWS Elemental MediaLive

Go through the list of outputs that you created and set up the captions in each output group, one by one.

Follow the procedure that applies to the format category of the captions output:

- the section called “Embedded or object captions encodes”
- the section called “Sidecar or SMPTE-TT captions encodes”

Create embedded or object captions encodes

Follow this procedure if the format of the captions asset that you want to add belongs to the category of embedded, burn-in, or object. You set up the captions and video and audio in the same output.

1. In the **Create channel** or **Edit channel** page for the channel, in the **Channel** panel, find the output group where you want to set up captions.
2. If you have already set up outputs in this output group with video (and possibly audio), find the outputs where you want to add the captions. Or create a new output in this output group.
3. In the output, go to the **Stream settings** and choose **Add caption**, then **Create a new caption encode**. Captions fields appear..
4. Complete the following fields:
   - **Captions description name**: Enter a name that is unique in the channel, for example, **`Embedded`**.
   - **Captions selector name**: Select the captions selector that you created when you created the captions selectors in the input. Specify the selector that identifies the captions asset that is the source for the captions in this output.
   - **Captions settings**: Choose the captions format for the output captions. Depending on the format, more fields appear.
5. Choose **Additional settings**. More fields appear. See the table after this procedure for information about which fields to complete for each format.

Complete the fields that appear for the selected format. For details about a field, choose the Info link beside the field.

6. You now have a captions encode that is fully defined. Repeat these steps to create more captions in this output or in another output, or in another output group.

Field	Topic	Applicable formats	For more information, see this section
Font, Positioning, Font Style	Captions style	Burn-in, DVB-Sub	the section called "Font styles for Burn-in or DVB-Sub"
Language code, Language description	Language information for this specific caption	All formats	Optional. For information, choose the **Info** link next to each field.
Accessibility, Caption DASH Roles, DVB DASH accessibility	Accessibility data	All formats	the section called "Including accessibility data in captions"
PIDs	PID assignment	ARIB, DVB-Sub	the section called "PIDs for ARIB", the section called "PIDs for DVB-Sub "
Captions language mappings	Tags in manifest	HLS	the section called "Language information in HLS manifests"

Create sidecar or SMPTE-TT captions encodes

Follow this procedure if the format of the captions asset that you want to add is a sidecar, or if the format is SMPTE-TT for a Microsoft Smooth output group.

You set up the captions and video in the same output.

1. In the **Create channel** or **Edit channel** page for the channel, in the **Channel** panel, find the output group where you want to set up captions.
2. Create a new output in this output group.
3. In the output, go to the **Stream settings** and choose **Add caption**, then **Create a new caption encode**. Captions fields appear.
4. Complete the following fields:
   - **Captions description name**: Enter a name that is unique in the channel, for example, **`Embedded`**.
   - **Captions selector name**: Select the captions selector that you created when you created the captions selectors in the input. Specify the selector that identifies the captions asset that is the source for the captions in this output.
   - **Captions settings**: Choose the captions format for the output captions. Depending on the format, more fields appear.
5. Choose **Additional settings**. More fields appear. See the table after this procedure for information about which fields to complete for each format.
6. You now have a captions encode that is fully defined. Repeat these steps to create more captions in this output group.

Field	Topic	Applicable formats	Description
Style Control, Fill Line Gap, Font Family, Copyright Holder	Captions style	EBU-TT-D	See the section called "Font styles for EBU-TT-D"
Style Control	Captions style	TTML, Web-VTT	See the section called "Font styles for TTML" or the section called "Font styles for WebVTT"
Language code, Language description	Language informati on for this specific caption	All formats	Optional. For information, choose the **Info** link next to each field.

Field	Topic	Applicable formats	Description
Accessibility, Caption DASH Roles, DVB DASH accessibility	Accessibility data	All formats	the section called "Including accessibility data in captions"
PIDs	PID assignment	Teletext	the section called "PIDs for Teletext ",

Including accessibility data in captions

In the captions in CMAF Ingest, HLS, MediaPackage, or Microsoft Smooth output groups, you can include accessibility data. This data describes the type of accessibility that the encode represents. For example, a captions track might provide a written translation (into another language) of the speech in the content. Accessibility data is also known as accessibility signaling.

Topics

- Supported accessibility data standards
- Specifying data in a CMAF Ingest or Microsoft Smooth output
- Specifying data in an HLS or MediaPackage output

Supported accessibility data standards

MediaLive supports the following styles of accessibility data.

Accessibility data style	Specification	CMAF Ingest	HLS or MediaPackage	Microsoft Smooth
DASH role captions	DASH role scheme (*ISO/IEC 23009-1:2022(E)*)	Yes		Yes
DVB DASH accessibility	*ETSI TS 103 285 Technical*	Yes		Yes

Accessibility data style	Specification	CMAF Ingest	HLS or MediaPackage	Microsoft Smooth
	Specification, V1.3.1 (2020-02)			
Accessibility	Signaled in tags that are inserted in the HLS manifest.		Yes	

Specifying data in a CMAF Ingest or Microsoft Smooth output

You can set up the captions encode to include accessibility data when you create the encode, as described in the section called "Embedded or object captions encodes" and the section called "Sidecar or SMPTE-TT captions encodes".

In the output that has the captions encode that you want to set up, follow these steps:

- To include DASH Roles, choose **Add dash role**s as many times as you want. Choose the style in each role.
- To include DVB DASH accessibility style, in **DVB DASH accessibility**, choose the applicable description. You can add only one instance of this accessibility style.

You can add more than one style of accessibility data to each encode. For example, you can add Dash Roles and DVB DASH accessibility style. You might want to do this because different downstream systems for these outputs implement different styles.

Handling of accessibility data in CMAF Ingest or Microsoft Smooth

The fields for accessibility data appear for all output group types, including types that don't support this data.

> **Note**
>
> When you set up audio encodes and you plan to include accessibility data, proceed as follows. First create the audio encodes in the CMAF Ingest and/or Microsoft Smooth output

groups, and set up the accessibility data. Then create the audio encodes in the other output groups.

Handling in supported output groups

If you aren't implementing shared captions encodes, MediaLive includes the data only in the captions outputs of the CMAF Ingest and Microsoft Smooth output groups that you set up for captions accessibility data.

Handling in shared encodes

You might plan to share captions encodes among several output groups. For example, you might share a captions encode among one CMAF Ingest output group and other output groups.

If you set up accessibility data in a shared audio encode, MediaLive will handle the data as follows:

- It will include the data in the CMAF Ingest and Microsoft Smooth output groups that share the encode.
- It won't include the data in other output groups, because those output groups don't support this data. Even though the output group is sharing the encode, MediaLive won't include the data.

Handling in other output groups

You might try to set up accessibility fields in an output that doesn't support accessibility data. If you're not implementing encode sharing with a CMAF Ingest or Microsoft Smooth output group, you will get an error message when you save the channel.

Specifying data in an HLS or MediaPackage output

You can set up the captions encode to include accessibility data when you create the encode, as described in the section called “Embedded or object captions encodes”.

In the output that has the captions encode that you want to set up, in **Accessibility**, choose **IMPLEMENTS_ACCESSIBILITY_FEATURES**.

MediaLive assigns accessibility captions a unique attribute in the EXT-X-MEDIA tag of the HLS manifest:

```
CHARACTERISTICS="public.accessibility.describes-spoken-
dialog,public.accessibility.describes-music-and-sound"
```

Here is an example of the EXT-X-MEDIA tag with the accessibility caption attribute:

```
#EXT-X-MEDIA:TYPE=SUBTITLES,GROUP-ID="captions-group",NAME="accessibility-
captions1",LANGUAGE="eng", CHARACTERISTICS="public.accessibility.describes-
spoken-dialog,public.accessibility.describes-music-and-
sound",AUTOSELECT=YES,DEFAULT=YES,URI="caption-accessibility-eng.m3u8"
```

Handling of accessibility data in HLS or MediaPackage output groups

The **Accessibility** field appears for all output group types, including types that don't support this data.

> **Note**
>
> When you set up audio encodes and you plan to include accessibility data, proceed as follows. First create the audio encodes in the HLS and/or MediaPackage output groups, and set up the accessibility data. Then create the audio encodes in the other output groups.

Handling in supported output groups

If you aren't implementing shared audio encodes, MediaLive includes the data only in the audio outputs of the HLS and MediaPackage output groups that you set up for audio accessibility data.

Handling in shared encodes

You might plan to share captions encodes among several output groups. For example, you might share a captions codec among one HLS output group and other output groups.

If you set up accessibility data in a shared captions encode, MediaLive will handle the data as follows:

- It will include the data in the HLS and MediaPackage output groups that share the encode.
- It won't include the data in other output groups, because those output groups don't support this data. Even though the output group is sharing the encode, MediaLive won't include the data.

Handling in other output groups

You might try to set up **Accessibility** in an output that doesn't support accessibility data. If you're not implementing encode sharing with an HLS or MediaPackage output group, you will get an error message when you save the channel.

Details for specific output formats

Following is information that applies only to the specified captions format.

Font styles for Burn-in or DVB-Sub

If you are setting up Burn-in or DVB-Sub captions, you might decide to specify the look of the captions. The following rule applies.

If you are using the same captions source in several outputs and all those outputs use the same format, then you must set up the font style information identically in each output. If you don't, you get an error when you save the channel. For example, you have an Archive output that includes DVB-Sub captions converted from captions selector "embedded". And you have a UDP output that also includes DVB-Sub captions converted from the same captions selector.

Note that you must set up the font style information separately—in the Archive output, and then in the UDP output. But you must enter the same information in both outputs.

For example, output A might use **Captions Selector 1** with the **Destination Type** set to **Burn-in**. And output B might also use **Captions Selector 1** with the **Destination Type** set to **Burn-in**. You set the font information once in output 1 and again in output 2. But you must set up all the font information identically in both outputs.

PIDs for ARIB

If you are setting up ARIB captions in a UDP/TS output group, you must specify the output PID.

- In the relevant UDP output group, choose the output that has the ARIB captions.
- For **PID settings**, complete **ARIB captions PID control** and **ARIB captions PID** as shown in the following table.

ARIB Captions PID Control	ARIB Captions PID	Result
Auto	Ignore	A PID is automatically assigned during encoding. This value could be any number.

ARIB Captions PID Control	ARIB Captions PID	Result
Use Configured	Enter a decimal or hexadecimal	This PID is used for the captions.

PIDs for DVB-Sub

If you are setting up DVB-Sub captions in a UDP/TS output group, you must specify the output PID.

- In the relevant UDP output group, choose the output that has the DVB-Sub captions.
- For **PID settings**, in **DVB-Sub PIDs**, enter the PID for the DVB-Sub captions in this output. Or keep the default.

PIDs for Teletext

If you are setting up Teletext captions in a UDP/TS output group, you must specify the output PID.

- In the relevant UDP output group, choose the output that has the Teletext captions.
- For **PID settings**, in **DVB Teletext PID**, enter the PID for the Teletext captions in this output. Or keep the default.

Language information in HLS manifests

If you are setting up captions in an HLS or output group, you must include captions language information in the manifest.

If the captions are embedded captions and the output is HLS, you must include captions language information in the manifest. If you don't include this information, the downstream player won't have information about the embedded captions. To include language information in the manifest:

1. In the HLS output group in Output groups, go to the **Captions** section. In **Captions language setting**, choose **Insert**. Choosing this option inserts lines in the manifest for each embedded captions language. It inserts as many lines as the mappings that you will add in the next step.

> **Note**
>
> This **Captions** section is in the output group. Don't confuse this section with the the captions encode sections in the individual outputs.

2. Still in the HLS output group, for **HLS settings**, in **Captions language mappings**, choose **Add captions language mappings**.
3. Choose **Add captions language mappings** again to add more mapping groups, one for each embedded captions asset, to a maximum of four groups. For example, if the output embedded languages contain English, French, and Spanish, you need three mapping groups.
4. Complete each mapping group to identify the CC (caption channel) number and its language. Specify the language as a three-letter ISO language code, as per ISO 639-2. For example, if captions channel 1 is French, then set up the three fields with "1", "fre", and "French".

   The order in which you enter the languages must match the order of the captions in the source. For example, if the captions are in the order French, then English, then Spanish, then Portuguese, then set up CC1 as French, CC2 as English, and so on. If you don't order them correctly, the captions in the manifest will be tagged with the wrong languages.

Font styles for EBU-TT-D

If you are setting up EBU-TT-D captions from source captions that are embedded or Teletext captions, you can optionally specify some of the font style information.

An EBU-TT-D caption encode consists of an XML file that the downstream system reads and processes. This XML file includes a section for font style information. You can specify some of this information.

1. In the output that has the EBU-TT-D captions, display the section for the captions.
2. Complete these fields. For details about a field on the MediaLive console, choose the **Info** link next to the field.
   - **Style control**
   - **Fill line gap**
   - **Font family**

This setup results in one of the following options:

The XML file for the captions includes the following style information:

Style information	Value in XML file for Include option	Value in XML file for Exclude option
Font style information (position, alignment, italics, and so on)	Set to match the source captions.	Left blank.
Font color and background color	Set to match the source captions.	Set to white font and black background.
Font size	Set to 100%.	Set to 100%.
Font family	Set to the value that you specified in **Font family**.	Set to **monospaced**.
Line gap	Set up to match the value that you specified in **Fill line gap**.	Set up to leave the gap unfilled.

Font styles for TTML

If you are setting up TTML captions from source captions are embedded or Teletext captions, you can optionally specify some of the font style information.

1. In the output that has the TTML captions, display the section for the captions.
2. Set **Style control** to **Passthrough** or **Use_configured**.

   Note that when **User_configured** is selected, there are actually no fields that you can configure.

The XML file for the captions will include the following style information:

Style information	Value in XML file for Passthrough option	Value in XML file for User-configured option
Font style information (position, alignment, italics, and so on)	Set to match the source captions.	Left blank.
Font color and background color	Set to match the source captions.	Set to white font and black background.
Font size	Match size of source captions, if specified. Otherwise, set to 80% of the available height available for captions.	Left blank.
Font family	Match family of source captions, if specified. Otherwise, set to **monospace SansSerif**.	Left blank.
Line gap	Set to leave the line gap unfilled.	Set to leave the gap unfilled.

Font styles for WebVTT

If you are setting up WebVTT captions from source captions are embedded or Teletext captions, you can optionally pass through some of the style information.

1. In the output that has the WebVTT captions, display the section for the captions.
2. Set **Style control**:
   - **No_Style_Data**: Includes only text and timestamp information for the caption encode.
   - **Passthrough**: Passes through position and color style data from the source, and includes the text and timestamp information.

Examples of handling captions in AWS Elemental MediaLive

The following examples describe how to implement the use cases from the section called “Typical scenarios”.

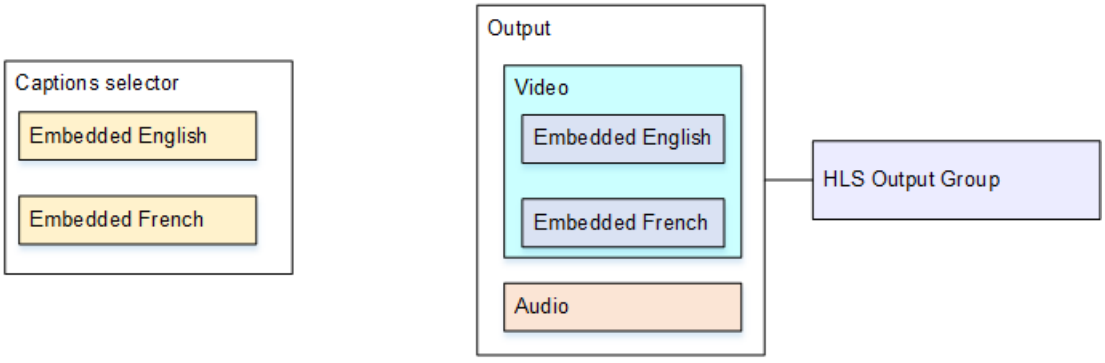
Topics

Use case A: One input format to one output and not converted

This example shows how to implement the first use case from the typical scenarios. The input is set up with one format of captions and two or more languages. Assume that you want to maintain the format in the output, and that you want to produce only one type of output and include all the languages in that output.

For example, the input has embedded captions in English and French. You want to produce an HLS output that includes embedded captions in both English and French, plus one video and one audio.

This example illustrates two important features of an embedded passthrough workflow. First, you don't create separate captions selectors; all the languages are automatically included. Second, if you are outputting to HLS, there is an opportunity to specify the languages and the order in which they appear.



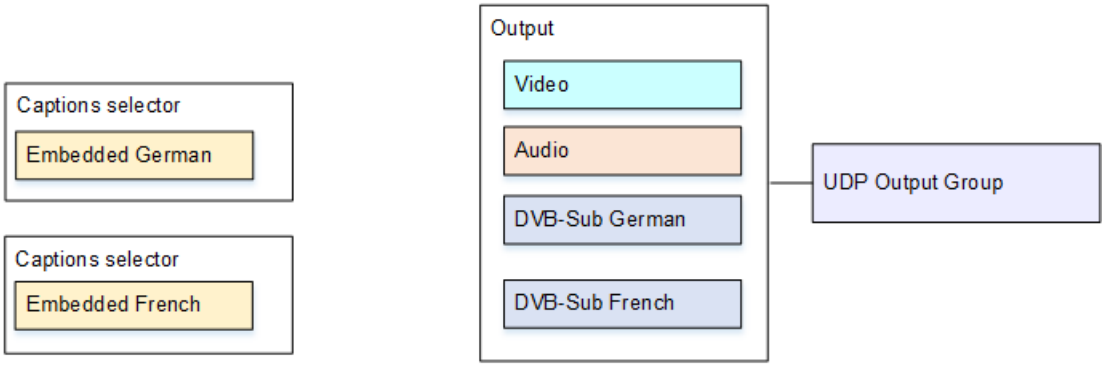
To set up for this use case, follow this procedure.

1. In the channel that you are creating, in the navigation pane, for **Input attachments**, choose the input.

2. For **General input settings**, choose **Add captions selector** to create one captions selector. Set **Selector settings** to **Embedded source**.
3. Create an HLS output group.
4. Create one output and set up the video and audio.
5. In that same output, create one captions asset with the following:
   - **Captions selector name**: Captions selector 1.
   - **Captions settings**: One of the Embedded formats.
   - **Language code** and **Language description**: Keep the field blank. With embedded captions, all the languages are included.
6. In the HLS output group, in **Captions**, for **Captions language setting**, choose **Insert**.
7. For **HLS settings**, in **Captions language mappings**, choose **Add captions language mappings** twice (once for each language).
8. Complete the first group of mapping fields with **1**, **ENG**, and **English** and the second group with **2**, **FRE**, and **French**.
9. Finish setting up the channel and save it.

Use case B: One input format converted to one different output format

This example shows how to implement the second use case from the typical scenarios. The input includes two captions languages, and the single output converts those captions. For example, the input has embedded captions in German and French. You want to produce a UDP output with both captions converted to DVB-Sub, plus one video and one audio.



To set up for this use case, follow this procedure.

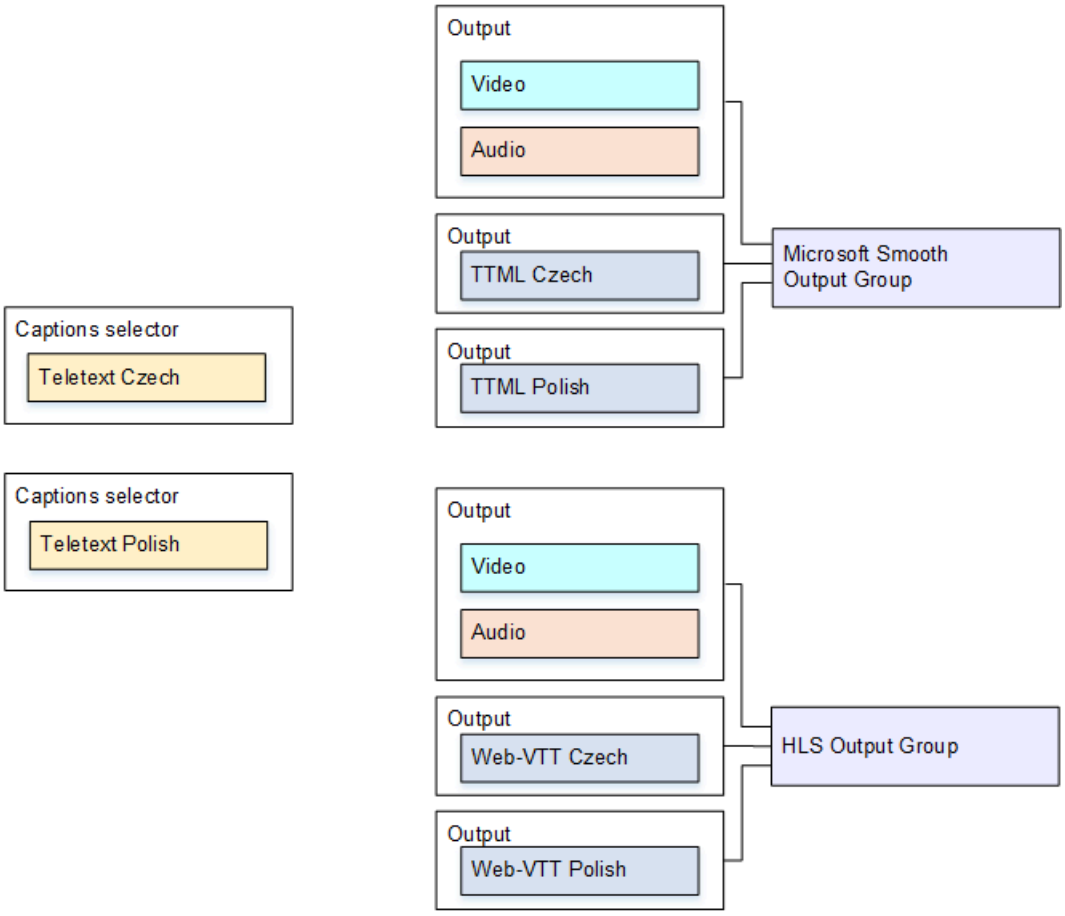
1. In the channel that you are creating, in the navigation pane, for **Input attachments**, choose the input.

2. For **General input settings**, choose **Add captions selector** twice, to create Captions selector 1 (for German) and Captions selector 2 (for French). In both cases, set **Selector settings** to **Embedded source**.
3. Create a UDP output group.
4. Create one output and set up the video and audio.
5. In this output, choose **Add captions** to create a captions encode.
   - **Captions selector name**: Captions selector 1.
   - **Captions settings**: DVB-Sub.
   - **Language code** and **Language description**: German.
   - Other fields: Keep the defaults or complete as desired.
6. Choose **Add captions** again to create another captions encode. Set up this encode for the French captions. Make sure that you set up the font fields for German and French in exactly the same way.
7. Finish setting up the channel and save it.

Use case C: One input format converted to different formats, one format for each output

This example shows how to implement the third use case from the typical scenarios. The input is set up with one format of captions and two or more languages. You want to produce several different types of output. In each output, you want to convert the captions to a different format but include all the languages.

For example, the input has Teletext captions in Czech and Polish. Assume that you want to produce a Microsoft Smooth output and an HLS output. Assume that in the Microsoft Smooth output, you want to include one video and one audio and you want to convert the captions to TTML. In the HLS output, you want to include one video and one audio and you want to convert the captions to WebVTT.



To set up for this use case, follow this procedure.

1. In the channel that you are creating, in the navigation pane, for **Input attachments**, choose the input.
2. For **General input settings**, choose **Add captions selector** twice to create the following captions selectors:
   - Captions selector 1 for Teletext Czech. Specify the page that holds the Czech captions.
   - Captions selector 2 for Teletext Polish. Specify the page that holds the Polish captions.

   Although you are including the captions in two different outputs (Microsoft Smooth and HLS), you need to extract them from the input only once, so you need to create only one captions selector for each language.
3. Create a Microsoft Smooth output group and configure it as follows:
   - Create one output and set up the video and audio.
   - Create a second output that contains one captions encode and no video or audio encodes, and with the following settings:

- **Captions selector name**: Captions Selector 1.
- **Captions settings**: TTML.
- **Language code** and **Language description**: Czech.
- **Style control**: Set as desired.

- Create a third output that contains one captions encode and no video or audio encodes, with the following settings:
  - **Captions selector name**: Captions Selector 2.
  - **Captions settings**: TTML.
  - **Language code** and **Language description**: Polish.
  - Other fields: same as the second output (the Czech captions).

4. Create an HLS output group and configure it as follows:

- Create one output and set up the video and audio.
- Create a second output that contains one captions encode and no video or audio encodes, and with the following settings:
  - **Captions selector name**: Captions Selector 1.
  - **Captions settings**: WebVTT.
  - **Language code** and **Language description**: Czech.
  - Other fields: Set as desired.
- Create a third captions output that contains one captions encode and no video or audio encodes, and with the following settings:
  - **Captions selector name**: Captions Selector 2.
  - **Captions settings**: WebVTT
  - **Language code** and **Language description**: Polish.
  - Other fields: same as the second output (the Czech captions).

5. Finish setting up the channel and save it.

Use case D: One captions output shared by multiple video encodes

This example shows how to set up captions in an ABR workflow.

The first setup shows how to set up an ABR workflow when the captions are in the same output as the video, meaning that the captions are either embedded or captions style.

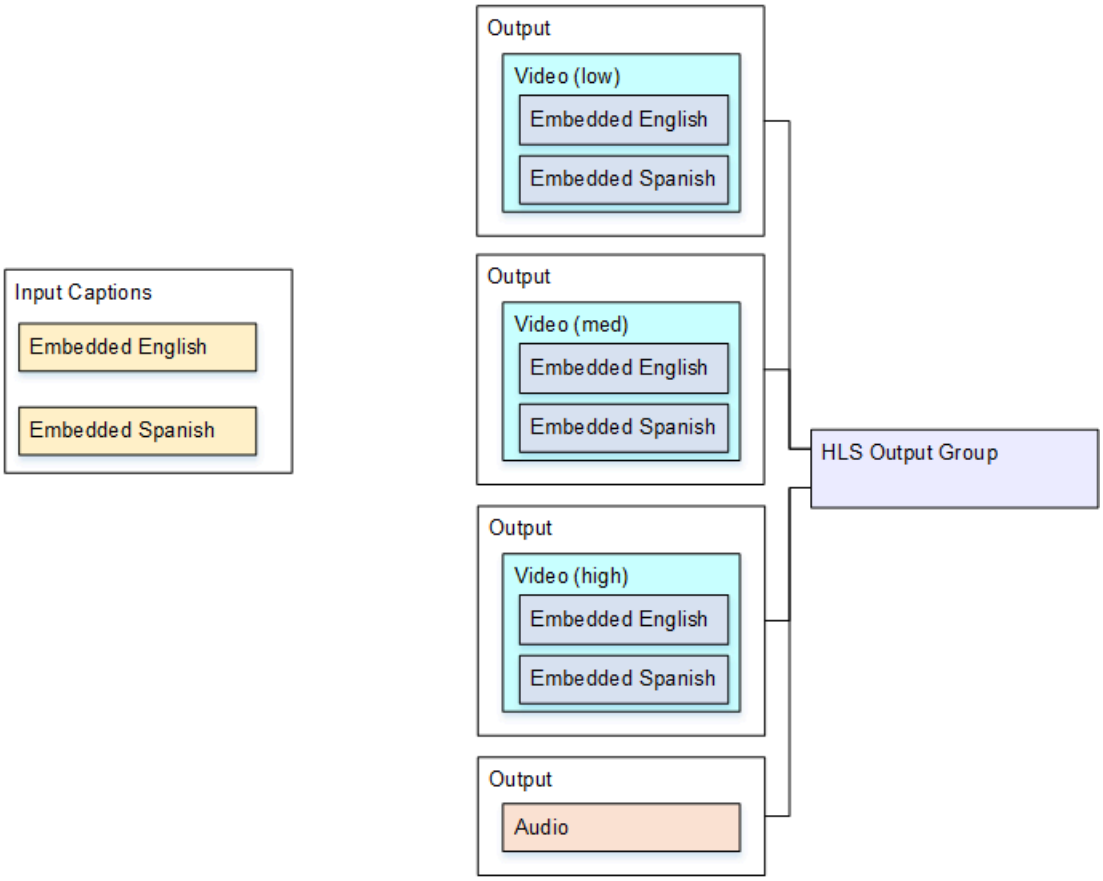
The second setup shows how to set up an ABR workflow when the captions belong to the sidecar category, in which case each captions encode is in its own output.

Topics

- Setup with Embedded or object-style captions
- Setup with sidecar captions

Setup with Embedded or object-style captions

This example shows how to implement the fourth use case from the typical scenarios. For example, you want to produce an HLS output with three video encodes (one for low-resolution video, one for medium, one for high) and one audio. You also want to include embedded captions (in English and Spanish) and associate them with all three video encodes.



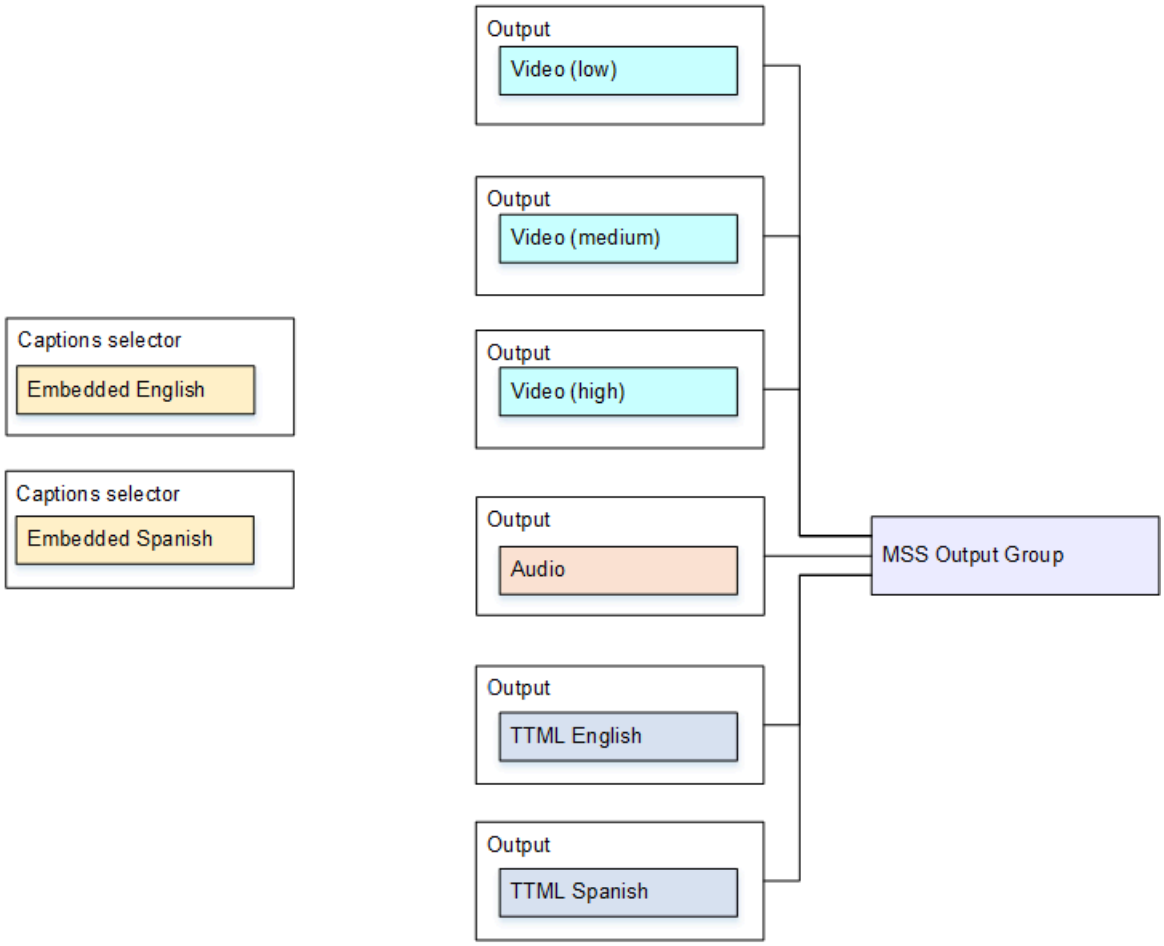
To set up for this use case, follow this procedure.

1. In the channel that you are creating, in the navigation pane, in **Input attachments**, choose the input.

2. For **General input settings**, choose **Add captions selector** to create one captions selector. Set **Selector settings** to **Embedded source**.
3. Create an HLS output group.
4. Create one output and set up the video and audio for low-resolution video.
5. In that same output, create one captions asset with the following:
   - **Captions selector name**: Captions selector 1.
   - **Captions settings**: One of the Embedded formats.
   - **Language code** and **Language description**: Leave blank; with embedded passthrough captions, all the languages are included.
6. Create a second output and set up the video and audio for medium-resolution video.
7. In that same output, create one captions asset with the following:
   - **Captions selector name**: Captions selector 1.
   - **Captions settings**: One of the Embedded formats.
   - **Language code** and **Language description**: Keep blank. With embedded captions, all the languages are included.
8. Create a third output and set up the video and audio for high-resolution video.
9. In that same output, create one captions asset with the following:
   - **Captions selector name**: Captions selector 1.
   - **Captions settings**: One of the Embedded formats.
   - **Language code** and **Language description**: Keep blank. With embedded captions, all the languages are included.
10. Finish setting up the channel and save it.

Setup with sidecar captions

This example shows an ABR workflow where the captions are in sidecars. For example, you want to produce a Microsoft Smooth output with three video encodes (one for low-resolution video, one for medium, one for high) and one audio. These encodes are in a Microsoft Smooth output. You want to ingest embedded captions (in English and Spanish) and convert them to TTML captions, one for English and one for Spanish.



To set up for this use case, follow this procedure.

1. In the channel that you are creating, in the navigation pane, for **Input attachments**, choose the input.
2. For **General input settings**, choose **Add captions selector** twice to create the following captions selectors:
   - Captions selector 1: for Embedded English.
   - Captions Selector 2: for Embedded Spanish.
3. Create a Microsoft Smooth output group.
4. Create one output that contains one video encode and set it up for low-resolution video.
5. Create a second output that contains one video encode and set it up for medium-resolution video.
6. Create a third output that contains one video encode and set it up for high-resolution video.
7. Create a fourth output that contains one audio encode and no video encode.

8. Create a fifth output that contains one captions encode and no video or audio encodes, and with the following settings for the captions encode:
   - **Captions selector name**: Captions selector 1.
   - **Captions settings**: TTML.
   - **Language code** and **Language description**: English.
9. Create a sixth output that contains one captions encode and no video or audio encodes, and with the following settings for the captions encode:
   - **Captions selector name**: Captions selector 2.
   - **Captions settings**: TTML.
   - **Language code** and **Language description**: Spanish.
10. Finish setting up the channel and save it.

Partner CDI inputs

A partner CDI input is a specific configuration of a CDI input. You must set up two CDI inputs as partners if you want to support automatic input failover for a CDI source. The two inputs always work together, as the two inputs in an automatic failover pair. The two inputs can be used only together, as a failover pair.

Topics

Regular inputs versus partner inputs

When you create a CDI input, you must decide whether you need to create a *regular* CDI input or a set of *partner* CDI inputs. This decision depends on how you want to implement pipeline redundancy and automatic input failover.

The following table describes the type of input to create depending on the workflow.

Channel is set up for pipeline redundancy	You want to set up this input for automatic input failover	Type of inputs to create
No (single-pipeline channel)	No	One regular CDI input.
	Yes	One set of partner CDI inputs —two CDI inputs set up as partners.
Yes (standard channel)	No	One regular CDI input.
	Yes	Two sets of partner inputs: • Two CDI inputs set up as one set of partner inputs. • Two more CDI inputs set up as another set of partner inputs.

Rules for using partner CDI inputs

These rules apply to partner inputs:

- Automatic failover – You can only use the partner inputs as a failover pair.
- Input switching – You can't use the partner inputs in an input switching workflow, where sometimes you switch to one partner and at other times you switch to the other partner.
- Single channel – You can use the partner inputs only in one channel. You can't attach one partner to one channel, and the other partner to a different channel.

Creating the set of partner inputs

To create the partner inputs, you must follow a special procedure. See the section called “CDI input – Partner CDI input”.

Editing the set of partner inputs

You can edit the inputs in the same way as you update regular CDI inputs. See the section called "Editing an input".

Deleting partner inputs

The two inputs have equal standing. The first input that you create when you follow the special procedure isn't the owner input or principal input. Therefore, these rules apply when you delete a partner input:

- You can delete one input without deleting the other.

  If you do so, the remaining input simply becomes a regular CDI input. If you delete the first input, the name of the second input doesn't automatically change. For example, if the input had the name myInput - partner, it will still have the name myInput - partner, even though it is no longer a partner CDI input. You can edit the input to change the name.
- You can delete the second input, then create the partner input again, from the first input. The IP addresses of the new input will be assigned the port 5001.
- You can delete the first input, then create the partner input again, from the second input. The IP addresses of the new input will be assigned the port 5000.

  If you didn't change the name of the second input (the default has the suffix, for example, **myInput - partner**), then the new input has the name **myInput - partner - partner**. You can edit the input to change the name.

Channel class and input class

One of the characteristics of a channel is its class. One of the characteristics of an input is its class. You set both the channel class and input class to implement or to omit pipeline redundancy.

Read this section for an overview of channel class and input class. Then for detailed information about implementing or omitting pipeline redundancy, see the section called "Pipeline redundancy".

About channel classes

When you plan the workflow, you must decide on the class for the channel. There are two channel classes:

- Standard class

  A *standard channel* has two encoding pipelines. When there are two pipelines, both pipelines perform the encoding. If one pipeline fails, output to the downstream system can continue, from the other pipeline. For more information and diagrams about exactly how MediaLive handles the failure, see the section called “Pipeline redundancy”.
- Single-pipeline class

  A *single-pipeline channel* has one encoding pipeline. If the single pipeline fails, output to the downstream system stops.

You set the channel class when you create the channel. You can upgrade or downgrade the class of an existing channel.

About input classes

As part of the steps for implementing or omitting pipeline redundancy in the channel, you must decide on the class for each input. There are two input classes:

- Standard class

  A standard-class input has two pipelines.

  All types of inputs can be set up as standard-class inputs.
- A single-class input has one pipeline.

  Not all inputs can be set up as single-class inputs. CDI inputs and RTP inputs can't be set up as single-class inputs.

Combinations of channel and input class

The following table summarizes the valid combinations of channel class and input class. The section the section called “Pipeline redundancy” provides information about choosing the appropriate combination for your workflow.

Channel	Inputs
Standard channel	All inputs must be standard-class inputs.

Channel	Inputs
Single-pipeline channel	The inputs might be a combination of single-class inputs and standard-class inputs. Most types of inputs will be single-class inputs. Any CDI inputs or RTP inputs will be standard-class inputs.
Single-pipeline channel	The inputs might all be standard-class inputs.

Dynamic inputs

You can set up a multiple-input channel with static and dynamic file inputs. You can then use the input switching feature of the channel schedule to switch from one input to another. A static input always connects to the same file. A dynamic input points to a different file each time it is used in an input switch in the schedule.

Using dynamic inputs lets you increase the number of video sources that you can use in the channel, while still observing the limit on the number of inputs that you can attach to the channel.

Only MP4 or Transport Stream (TS) file inputs that are stored in one of the following locations can be set up as dynamic inputs:

- Amazon S3
- AWS Elemental MediaStore

Setting up dynamic inputs

For information about planning and creating dynamic inputs, see the section called “Dynamic inputs”.

Working with ID3 metadata

You can include ID3 metadata in Archive outputs, HLS outputs, MediaPackage outputs, and UDP outputs. Typically, you include ID3 metadata in an output if you know that a downstream system expects the data and is capable of interpreting it.

You should obtain the requirements for ID3 metadata from a representative of the downstream system.

When you are creating or editing a channel, you can set up individual outputs in a channel so that ID3 metadata is enabled. The ID3 metadata can come from one or more of the following sources:

- It can be metadata that is already present in the input.
- It can be metadata that you add when you create the channel.
- It can be metadata that you add by creating actions in the schedule.

ID3 metadata is included in the Archive, HLS, MediaPackage, or UDP output according to the specific rules for the output type.

Topics

Enabling ID3 metadata

To include ID3 metadata in an output, you must enable ID3 metadata in that output when you create or edit the channel.

Enabling in archive outputs

To include ID3 metadata in Archive outputs, you must enable the feature in each applicable output.

To enable ID3 metadata in Archive outputs

1. On the **Create channel** page, in the **Output groups** section, in the **Archive** group, choose the output where you want to enable ID3 metadata.

2. For **Container Settings**, for **PID Settings**, for **Timed Metadata Behavior**, choose **PASSTHROUGH**.
3. For **Timed Metadata PIDs**, enter the PID where you want to insert the ID3 metadata.
4. Repeat for each applicable output.

For information about the results of enabling, see the section called "Results of enabling ID3 metadata" later in this section.

Enabling in HLS outputs

To include ID3 metadata in HLS outputs, you must enable the feature in each applicable output.

To enable ID3 metadata in HLS outputs

1. On the **Create channel** page, in the **Output groups** section, in the **HLS** group, choose the output where you want to enable ID3 metadata.
2. Make sure that **HLS Settings** is set to **Standard hls**. Only standard outputs can contain ID3 metadata. The **Audio-only outputs** option (which is the other option in this field), is used to set up audio rendition groups and can't contain this metadata.
3. For **PID Settings**, **Timed Metadata Behavior**, choose **PASSTHROUGH**.
4. For **Timed Metadata PIDs**, enter the PID where you want to insert the ID3 metadata.
5. Repeat for each applicable output.

For information about the results of enabling, see the section called "Results of enabling ID3 metadata" later in this section.

Enabling in MediaPackage outputs

To include ID3 metadata in MediaPackage outputs, you don't have to set up the output. MediaPackage outputs are automatically set up with this feature enabled.

For information about handling of ID3 metadata in MediaPackage outputs, see the section called "Results of enabling ID3 metadata" later in this chapter.

Enabling in UDP outputs

To include ID3 metadata in UDP outputs, you must enable the feature in each applicable output.

To enable ID3 metadata in UDP outputs

1. On the **Create channel** page, in the **Output groups** section, in the **UDP** group, choose the output where you want to enable ID3 metadata.
2. For **Network Settings**, **PID Settings**, **Timed Metadata Behavior**, choose **PASSTHROUGH**.
3. For **Timed Metadata PIDs**, enter the PID where you want to insert the metadata.
4. Repeat for each applicable output.

For information about the results of enabling, see the section called “Results of enabling ID3 metadata” later in this section.

Results of enabling ID3 metadata

Here are the results of enabling ID3 metadata in the channel:

- ID3 metadata other than type TDRL or PRIV that is present in the input is automatically included in the eligible outputs.
- ID3 metadata of type TDRL or PRIV that is present in the input is passed through to eligible outputs as follows:
  - If the frame doesn't have "Elemental Technologies" included in the wording, the metadata is passed through.
  - If the frame has "Elemental Technologies" included in the wording, the metadata is not passed through. The metadata isn't passed through because MediaLive assumes that the timestamp for this metadata has passed.
- ID3 metadata that you set up in the output group is inserted in those outputs where you enabled ID3 metadata, when you created the channel. For information about setting up ID3 metadata in the output group, see the section called “Inserting ID3 metadata when creating the channel”.
- ID3 metadata that you set up by creating an action in the MediaLive schedule is included in the eligible outputs. For information about setting up ID3 metadata in the schedule, see the section called “ Inserting ID3 metadata using the schedule”.

The eligibility of an output depends on the output group type, as shown in the following table.

Type of output group	ID3 metadata that is present in input	ID3 metadata that you specify when setting up the channel	ID3 metadata that you insert using the schedule
Archive	Passed through	Not included in output	Included in output
HLS	Passed through	Included in output	Included in output
MediaPackage	Passed through	Not included in output	Included in output
UDP	Passed through	Included in output	Not included in output

Passing through ID3 metadata

You can set up outputs so that ID3 metadata that is in the channel input is automatically passed through to the output. To pass through ID3 metadata, enable ID3 in the outputs. For information, see the section called “Enabling ID3 metadata”.

Inserting ID3 metadata when creating the channel

You can set up to insert ID3 metadata at a regular cadence (for example, every 10 seconds) into HLS or UDP outputs where you enabled ID3 metadata. You can't insert ID3 metadata into Archive or MediaPackage outputs.

To insert ID3 metadata when creating the channel

1. Make sure that you enabled ID3 metadata. For detailed information, see the section called “Enabling ID3 metadata”.
2. On the **Create channel** page, in the **Output groups** section, choose the HLS group or the UDP group. (You can't insert ID3 metadata in an Archive group or MediaPackage group.)
3. Choose **ID3**.
4. For **Timed metadata ID3 frame**, choose the ID3 frame type that you want to apply to the metadata.

Try to avoid using PRIV for metadata that you insert when creating the channel and for metadata from one of the other sources.

5. For **Timed metadata ID3 period**, enter the repeat interval for the ID3 metadata, in seconds.

   For a UDP output group, set any length. For an HLS output group, we recommend that you set the period (interval) to half the segment length. To verify the segment length, in the **HLS output group**, choose **Manifests and Segments**, and look at **Segment Length**.

When you start the channel, the first ID3 metadata is inserted shortly after the output starts and then at the specified interval for the lifetime of the channel.

The timestamp in the ID3 metadata is derived from the output timecode. It indicates the time at which the ID3 frame is inserted into the output, when the channel is running. The timestamp is in the format that you specified for the **Source** field in the **Timecode Configuration** section of the **General Settings** page for the channel.

Inserting ID3 metadata using the schedule

You can insert ID3 metadata at a specific time by creating an action in the MediaLive schedule. The metadata is inserted in each HLS output or MediaPackage output where you have enabled ID3 metadata. It is not inserted in UDP outputs.

Typically, you include ID3 metadata in accordance with instructions of the downstream system.

To insert ID3 metadata

1. Make sure that you enabled ID3 metadata. For detailed information, see the section called "Enabling ID3 metadata".
2. Create actions in the schedule. For detailed information, see *Setup: Creating a schedule*.

Working with ID3 segment tags

You can include ID3 tags in every segment in the HLS outputs and MediaPackage outputs in an AWS Elemental MediaLive channel. Typically, you include ID3 segment tags in an output if you know that a downstream system expects the data and can interpret it.

You should obtain the requirements for the contents of the tag from a representative of the downstream system.

How the feature works

You enable the feature. For an HLS output group, you set up individual HLS output groups in a channel so that ID3 segment tagging is enabled for all the outputs in the output group. For a MediaPackage output group, there is no setup. Tagging is always enabled in these output groups.

You then create an ID3 segment tag action in the channel schedule and specify the contents of the tag. There are two options for the contents:

- You provide the content for the Field in the ID3 tag portion of the ID3 frame. The Field is always type TXXX. MediaLive constructs the entire ID3 frame.

- You provide the entire ID3 frame.

At the start time for the action, the channel starts to insert the tag content in every segment in the HLS and MediaPackage outputs.

You can change the contents of the tag, by creating a new action. At the start time of the new action, MediaLive starts inserting the contents of the new tag in every segment.

Topics

Comparison to ID3 metadata

A feature that is similar to ID3 segment tagging is ID3 timed metadata. You can set up the channel to include both sets of metadata. Both sets are inserted in the same PID, but as different types of ID3 metadata.

Here is a comparison of the two features:

Topic	ID3 segment tags that uses the tag option	ID3 segment tags that uses the ID3 option	ID3 timed metadata
One time or repeat?	Repeat	Repeat	One-time

Topic	ID3 segment tags that uses the tag option	ID3 segment tags that uses the ID3 option	ID3 timed metadata
	At the start time of the action, MediaLive starts inserting ID3 tags in every segment in the applicable outputs. It continues to insert in every segment, typically for the life of the channel.	At the start time of the action, MediaLive starts inserting ID3 tags in every segment in the applicable outputs. It continues to insert in every segment, typically for the life of the channel.	At the start time of the action, MediaLive inserts the ID3 metadata in the applicable outputs, as a one-time event.
Applicable output groups	HLS and MediaPack age output groups. You must enable the ID3 segment tag feature in the applicable output groups.	HLS and MediaPack age output groups. You must enable the ID3 segment tag feature in the applicable output groups.	Archive, HLS, MediaPackage, and UDP output groups. You must enable the ID3 timed metadata feature in the applicable output groups.
You provide clear text or base64?	Clear text You provide only the value for the TXXX field inside the ID3 tag. You specify this value as clear text.	Base64 You provide a fully formed ID3 metadata item (including both a header and a frame, as per the ID3 specification) and encode it as base64.	Base64 You provide a fully formed ID3 metadata item (including both a header and a frame, as per the ID3 specification) and encode it as base64.

Topic	ID3 segment tags that uses the tag option	ID3 segment tags that uses the ID3 option	ID3 timed metadata
Typical contents	Content that you want to repeat in every segment Typically, the content consists in all or part of variable text in the form of MediaLive variable data. For example, it might consist of the date and time, and the current segment number, meaning that the tag contents are different in each segment.	Content that you want to repeat in every segment	Content that you want to include once

Inserting ID3 segment tags

To include ID3 segment tags in the outputs in an output group, enable ID3 segment tagging when you create or edit the channel. You don't need to enable the feature in MediaPackage output groups.

Then to set up the channel to start inserting the tag, create an action in the MediaLive schedule. The channel inserts the tag in the applicable HLS output groups and in every MediaPackage output group.

To enable ID3 segment tagging in HLS outputs

1. On the **Create channel** page, in the **Output groups** section, in the **HLS** group, choose **ID3**.
2. For **HLS ID3 Segment Tagging**, choose **ENABLED**.

3. Optionally, in each output, specify the PID for the tags. If you don't specify a PID in an output, MediaLive uses PID 502.

   Choose the output in this output group. For **Container Settings**, for **PID Settings**, for **Timed Metadata PIDs**, enter the PID where you want to insert the ID3 tag.

   Note that the ID3 timed metadata feature also uses this PID.
4. Repeat for each applicable output in the output group.

To enable ID3 segment tagging in MediaPackage outputs

MediaPackage output groups are automatically set up with ID3 segment tagging enabled, and with PID 502 specified.

Therefore, if you create an action in the schedule to insert tags, the MediaPackage outputs include that tag. If you don't create an action, the outputs don't include tags. (MediaLive doesn't insert a default tag.)

To start inserting ID3 segment tags

1. Make sure that you have enabled ID3 segment tags in the HLS output group.
2. Create an action in the schedule. For detailed information, see *Setup: Creating a schedule*.

   Typically, you create only one action in the schedule. If you create another action, the tag specified in that action replaces the tag specified in the previous action.

   To stop MediaLive inserting the tag, create an action with empty content.

Working with image overlays

You can impose static images onto a video in an MediaLive channel. A static image is a still image that doesn't have motion. You prepare the image or images and store them outside of MediaLive. You then use the schedule feature in MediaLive to set up a timetable that specifies when images will be inserted in the running channel, and when each will be removed.

Topics

- Step 2: Handling encode sharing
- Step 3: Inserting and removing an overlay

Two options: global overlay and per-output overlay

There are two options for inserting and removing image overlays in a channel — the global option and the per-output option.

- Global image overlay: Insert a static image overlay in every output in every output group.

- Per-outputs image overlay: Insert a static image overlay into the running channel, only in specific outputs in specific output groups. The action is called .

In one channel, you can combine the global action and the per-output option in the same channel. For example, you can insert image X globally (in all outputs) and insert image Y only in output A. Output A will have both image X and image Y. All other outputs will have only image X.

Topics

- How MediaLive handles a global image
- How MediaLive handles a per-output image
- Image layers and inserting images
- Removing images
- Properties of the image

How MediaLive handles a global image

MediaLive inserts the image before it sets the resolution in the video. In this way, the image gets resized as MediaLive resizes the video frame to obtain the specified resolution. The output image and video frame can resized to be smaller (a lower resolution) or larger (a higher resolution).

The global option works well when you want to use the same image in every video output, and you want that image to take up the same proportion of the video frame in every video output. When you prepare the image, you make sure that it takes up the desired proportion of the source video. For example, you might want its height to be 10% of the height of the underlying video. After MediaLive sets the video resolution, the image is still the same proportion of the underlying video.

For example, the image takes up approximately 10% of the height of an output 720p video frame and approximately 10% of the height of an output 4K video frame.

How MediaLive handles a per-output image

MediaLive sets the resolution of the video, and then it overlays the image. This means that the image doesn't get resized, it retains its absolute size.

The per-output option is useful if you want to overlay different images in different outputs. For example, you might want to insert one logo in the video in one output group, and another logo in the video in another output group.

The per-output option is also useful if you want to insert an image without resizing it. For example, if you want the image to have the same absolute size in every output in an ABR stack. The image is the same absolute size on a 720p video frame as it is on a 4K video frame. Therefore, you can use the same image file in all the outputs.

You might also want the image to have the same relative size in a group of outputs that have have different resolutions. For example, you want every image to take up 10% of the height. In this case, you must prepare separate files for each output, and create separate insert actions in the different outputs.

Image layers and inserting images

An image always exists in a layer. There are 8 global layers for the global option, and 8 per-output layers for the per-output option. A layer can contain only one image.

The layers are ordered. Layer 0 is at the bottom and layer 7 at the top.

The per-output layers are all on top of the global layers. This means that from the bottom, the layers are global layers 0 to 7, then per-output layers 0 to 7. Keep this layer order in mind if you plan to overlap images.

Removing images

There are two actions to deactivate (remove) an image, one to remove from a global layer, and one to remove from the per-outputs layer in specific outputs.

The global action removes the image from the specified layer and from all outputs.

The per-output action is more flexible. For example, you might insert image X in per-output layer 4 in outputs A and B. You might then insert image Y in per-output layer 4 in output C. You can

then enter a deactivate action that removes the image from per-output layer 4 in outputs A and C. Image X in output A will be removed, and image Y in output C will be removed. Image X in output B will still exist.

Properties of the image

Start time and duration

You can configure each image overlay with a start time and duration.

Positioning

You can insert the image overlay at any position on the video frame, relative to the X axis and Y axis of the video frame. You can position images so that they overlap each other.

Opacity and fade

You can configure with an opacity and with fade-in and fade-out.

Input insertion and overlays

You might insert image overlays in a channel where you are also performing input switching (to ingest different inputs). Keep in mind that the handling for input switches and image overlays is completely decoupled. In other words, you don't have to worry that when MediaLive switches to a different input, the currently active image overlays will disappear. They won't disappear.

Step 1: Preparing the static image overlay file

You must prepare each image overlay that you want to use in your channel, and store it in a suitable location, such as an Amazon S3 bucket. You can prepare the images at any time, either before you start the channel, or while the channel is running.

To prepare the overlay file

1. Determine the size (width and height in pixels) of the file you need. You might need several instances of one image, each in a different size. For more information, see the guidelines after this procedure.
2. Create files with the following characteristics:

   - 32-bit bmp, png, or tga format

- If you use a graphics program that outputs channels, set up to output the alpha channel. This ensures that the image overlay doesn't appear in a black or white box.

3. Place the prepared file in a location that is accessible to the MediaLive. Make a note of the location and of any user credentials that users need to access the file. You can specify the location in one of these ways:

   - Amazon S3 bucket, using SSL. For example:

     ```
     s3ssl://DOC-EXAMPLE-BUCKET/company-overlays/overlay.png
     ```

     With MediaLive, the Amazon S3 bucket name mustn't use dot notation. For example, `company-overlays` is acceptable but `company.overlays` isn't.
   - A location that supports HTTP or HTTPS. For example:

     ```
     https://203.0.113.0/corporate-logos/large.bmp
     ```

Determining the image size if you are using the global option

Keep in mind that with the global insertion option, MediaLive inserts the image on the output video frame *before* it sets the output video resolution. This means that the image will be resized with the output video.

Follow these guidelines:

- Determine the size of the image relative to the *source* video. For example, you might want an image to take up 10% of a 1280×720 source video frame. In this case, the image height should be approximately 72 pixels.
- You can prepare a new file that is the desired size. Or you can use an existing file and resize it when you prepare the insert action. MediaLive resizes the image before overlaying it on the video. Keep in mind that resizing might decrease the quality.
- If the channel has sources with different resolutions, you have two options:
  - You can optimize the image for one source.
  - Or you can create multiple versions of the same file, with each file in a different size. When you create an action to switch to a different input (with a different resolution), create a new insert image action, to insert the image that has the appropriate size.
- If the image is bigger than the source video frame, MediaLive trims off the excess.

Determining the image size if you are using the per-output option

Keep in mind that with the per-output insertion option, MediaLive inserts the image on the output video frame *after* it sets the output video resolution. This means that the image will be resized with the output video.

Follow these guidelines:

- Determine how many sizes you need of each image. For example, if you will insert the image on outputs that have three different resolutions, you need three different sizes.

  You can obtain the different sizes in either of these ways:
  - You can create multiple versions of the same file, with each file in a different size. Create separate actions for each size. In each action, specify all the outputs where the image will be inserted.
  - You can resize the image when you create the insert action. Create separate actions for each size. In each action, resize the image by setting a height and width. Specify all the outputs where the resized image will be inserted.
- If the image is bigger than the output video frame, MediaLive trims off the excess.

Step 2: Handling encode sharing

Read this section if you plan to use the per-output option, and you have already set up output groups to use video encode sharing. Video encode sharing involves creating one video encode, and then sharing it among two or more outputs in the same channel. For example, you might use the same video encode in the outputs in an ABR stack in an HLS output group and in an ABR stack in a Microsoft Smooth output group.

Video encode sharing isn't compatible with per-output image insertion. To undo sharing, follow this procedure:

1. Identify the encodes that you have set up for sharing:
   - In the **Create channel** or **Edit channel** page for the channel, find one of the output groups where you plan to insert per-output images. Select the first video output, then in **Stream settings**, select the video encode. If this video encode is shared, a note appears listing the other outputs.
   - Repeat for every output group in the channel. Make a list of the encodes and how they are shared.

2. If you plan different images for these outputs, you must stop sharing them:
   - Stop sharing the video encode in one of the outputs, for example in output group A. For instructions, see the section called “To stop sharing an encode”. The video encode is now used only in output group B.
   - In output group B, clone the video encode that was previously shared. Keep in mind that cloning isn't the same as sharing. For more information, see the section called “Creating by cloning”.

Step 3: Inserting and removing an overlay

When you are ready, you can create an action in the channel schedule to activate (insert) the overlay. You can create the action at any time – before the channel has started or while it is already running.The schedule is a timetable that is attached to each channel. It lets you perform actions at a specific time, on a running (active) channel. You can work with the schedule using either the MediaLive console or an AWS API or SDK.

You can set up the action so that an image overlay is active for a specific time, or so that it is active indefinitely. In both cases, you can stop the overlay at any time by creating a deactivate action. For more information, see the section called “Image overlays”.

Input clipping

You can clip a file input so that MediaLive ingests only a portion of the file. The file must be an MP4 file that is stored on Amazon S3, AWS Elemental MediaStore, or an HTTP server that supports HTTP range requests.

You clip a file as part of setting up an input switching action in the channel schedule. Therefore, to use a clipped file, you must use the schedule.

The integration with input switching works as follows. When MediaLive is getting ready to switch to the file input that includes input clipping, MediaLive sends a request to the upstream system, to request a portion of the file, rather than the entire file.

To set up a file input for input clipping

1. If the upstream system is an HTTP server, confirm with that system that they support range requests. If the server doesn't support range requests, there will be an input loss problem when the input switch occurs.

2. Create the MP4 file input in the usual way. See the section called “MP4 input”.
3. Attach the input to the channel in the usual way. See the section called “Inputs part 1: Attach inputs”.
4. Create a switch input action in the schedule that specifies the start time and end time for the clip. See the section called “Creating actions”.

   You can specify a start point (if you don't specify one, the ingest starts at the beginning of the file). You can specify an endpoint (if you don't specify one, the ingest stops at the end of the file). Or you can specify both a start point and end point.

When the channel switches to this input, it starts and stops ingesting the file at the specified points.

You can reuse this same input repeatedly, each time specifying a different portion to ingest. To do so, create another switch input action, with different start and end times.

Handling loss of video input

You can customize the way that MediaLive handles media when the video input into the channel is lost.

Topics

- How MediaLive handles video input loss
- Configuring the replacement content
- Customizing delivery

How MediaLive handles video input loss

When MediaLive is ingesting an input, it might detect that the video source has been lost. This loss causes MediaLive to start to follow the *input loss behavior* handling. MediaLive starts to encode *replacement content* (fill frames) on the output side. This handling ensures that the channel can continue to encode video content. (A key rule of MediaLive is that a running channel must always be encoding content.)

The channel follows the input loss handling until it recovers and goes back to normal encoding. The way that the channel recovers depends on whether you implement automatic input failover:

- If you implement automatic input failover, the input loss handling will continue until the video black failover condition triggers the switch to another input (or until the input recovers.) If the second input fails, input loss handling will restart and continue until you have fixed the problem with both of the inputs.
- If you don't implement automatic input failover, the input loss handling will continue until the input recovers or until you resolve the problem with the input.

The two features complement each other but work on different timing:

- Input loss handling occurs as soon as an expected frame fails to arrive. For example, if the framerate of the input is 60 FPS, the handling will be triggered if a frame does not arrive within 17 Msecs of the previous frame. (17 Msecs is approximately 1 second divided by 60.)
- The trigger for automatic input failover is longer and is configurable. A typical trigger is 1000 Msecs.

Input loss compared to input probing failure

Input loss handling occurs only after a previously healthy input becomes unhealthy.

It is also possible for an input to fail before that. When a channel starts and MediaLive begins to ingest the first input, it *probes* for the input—it attempts to detect the input and the sources. If the detection fails, then the input and the chanel fails immediately. You must resolve the problem and restart the channel. The problem might be that the input isn't present (this problem applies mostly to RTMP inputs), or the input exceeds the current channel specifications, or the input settings are wrong.

Default behaviour input loss handling

The default for input loss handling is the following:

- Encoding the replacement content: Repeat and encode the last valid frame that was received. Repeat for 1000 Msecs. Then encode black frames for 1000 Msecs. Then encode a black slate indefinitely.
- Delivering the content: The default handling is to emit (deliver) the encoded replacement content.

Customizing input loss handling

- You can customize the timing of the replacement content, and you can customize the content of the slate..
- In some output group types, you can change the delivery so that the encoded content isn't delivered.

Configuring the replacement content

You can customize the duration of the replacement content, and you can customize the image or color used for the slate. For example, you can change the slate to an image (such as *Please stand by*).

> **Note**
>
> This section assumes that you are familiar with creating or editing a channel, as described in *Setup: Creating a channel* .

1. On the **Create** channel page of the MediaLive console, choose **General settings**. Expand **Global configuration**. Choose **Enable global configuration**, if necessary.
2. In **Input loss behavior**, choose **Input loss Behavior**. More fields appear. These fields control the replacement content, as follows:
   - Encode the last valid frame for the time specified in **Repeat Frame Msec**. Zero means disabled (skip the repeat frame and go to the black frame). The value 1,000,000 means repeat the previous forever.
   - When **Repeat Frame Msec** expires, encode a black frame for the time specified in **Black Frame Msec**. Zero means disabled (skip the black frame and go to the slate). The value 1,000,000 means repeat black frames forever.
   - When **Black Frame Msec** expires, switch to sending a specified slate or color, as specified in **Input Loss Image Type** and then **Input Loss Image Color** or **Input Loss Image Slate**.
3. Complete one or more fields to customize the behavior. For details about a field on the MediaLive console, choose the **Info** link next to the field.

Customizing delivery

You can change the default handling of the replacement content so that instead of delivering the encoded output, MediaLivediscards it. You can change the handling in the following types of output groups:

- HLS
- Microsoft Smooth
- RTMP
- UDP/TS

For all other types of output groups except MediaPackage, MediaLive always delivers the content. For a MediaPackage output group, MediaLive always pauses delivery.

> **Note**
>
> This section assumes that you are familiar with creating or editing a channel, as described in *Setup: Creating a channel* .

1. On the **Create** channel page of the MediaLive console, choose the output group in the left navigation bar.
2. In the **Settings** section for the output group, find the **Input Loss Action** field for that output group. Choose the option you want for all the outputs (including outputs that don't include video) in this output group. See the table after this step.

This table lists the delivery options for the output groups. Read across each row.

Type of output group	Field	Description
HLS Microsoft Smooth RTMP	EMIT_OUTPUT	Deliver the replacement content. This is the default for these output groups.

Type of output group	Field	Description
HLS Microsoft Smooth RTMP	PAUSE_OUTPUT	Encode only the repeat frames, if the channel is configured to process them. After that content ends, stop delivery for all outputs in this output group. Note that MediaLive keeps the underlying RTMP connection open.
UDP	EMIT_PROGRAM	Encode the replacement content, and deliver the program and all the tables for this output group. This is the default for UDP.
UDP	DROP_TS	Stop delivery of the entire transport stream in this output group.
UDP	DROP_PROGRAM	Drop the program from the transport stream. MediaLive replaces the program with null packets, in order to meet the TS bitrate requirement. Deliver the null packets and all the tables for this output group.

Recommendation

You should make sure that the delivery meets the expectations of the downstream system.

For example, if the channel is a standard channel (with two redundant pipelines), the downstream system might be set up to switch to the output from the second pipeline. In this case, it's best if you set up the output to stop emitting the output.

As another example, the channel might have only one pipeline. Furthermore, the downstream system might not behave well if it loses delivery from MediaLive. Therefore, it's best for you to set up to emit the output. The downstream system will remain stable, and you could set up MediaLive with a "Please stand by" slate to improve the experience for the person watching the video.

Preparing inputs in AWS Elemental MediaLive

You can prepare an input that is associated with an immediate input switch in order to reduce the delay that occurs when MediaLive performs the switch.

If you prepare an input, there is much less delay when MediaLive performs an immediate input switch. This is because MediaLive has already probed the input and started to decode. If you don't prepare the input, there is a delay between the moment that the MediaLive schedule receives the action and the moment that the switch occurs.

We recommend that you prepare an input in this situation

- You plan to switch to an input with an immediate start type.
- You don't know when the switch will need to occur, but you do know that you might have only a few seconds advance notice.

You prepare an input by adding an *input prepare* action to the channel schedule. Typically, the input switch that the input prepare applies to is an immediate input switch. The input prepare itself can be set up to start at a fixed time, to start immediately, or to start following a specified input switch.

MediaLive adds the action to the schedule. At the action start time, MediaLive starts to prepare the input.

Note that there is no advantage to preparing an input if you will switch to it as a fixed input switch or a follow input switch. In this case, MediaLive automatically prepares the input ahead of time.

Terminology

In this section, we use the following terms:

- *Prepare action* – The input prepare action in the schedule.
- *Associated switch action* – The input switch action that the input prepare action is associated with. The prepare action prepares input A. The associated switch action switches to input A.
- *Fixed prepare* – An input prepare action that is set up to start at a fixed time.
- *Immediate prepare* – An input prepare action that is set up to start immediately.
- *Follow prepare, follow-start prepare, follow-end prepare* – An input prepare action that is set up to follow an input switch. The follow prepare can follow the start or the end of the referenced switch.
- *Reference switch action* – The input switch action that is being used as the trigger for a follow input prepare. So a follow input prepare follows the reference input switch action.

> **Note**
>
> This content in this section assumes that you are familiar with input switching, as described in the section called "Input switching".

Topics

Rules and limits for input prepare

One active prepare at a time

The schedule can contain any number of input prepare actions, but only one input prepare action can be active at one time.

Start time at least 10 seconds in advance

Set up each input prepare action so that it starts at least 10 seconds before the associated switch.

No RTMP pull inputs

A channel cannot have both RTMP pull input and the input prepare feature enabled. (RTMP push inputs are acceptable.) You must choose which feature is more important—the input prepare or the RTMP pull input.

- If you want to use the input prepare feature and the channel already has an RTMP pull input, you must first remove the input.
- If you want to add an RTMP pull input and the channel already has input prepare actions in the schedule, see the section called "Enabling the feature".

Setting up input prepare actions in the schedule

Follow this procedure to add input prepare actions to the channel schedule, in order to prepare any input ahead of the switch action to that input.

To include input prepare actions in a channel schedule

1. As a one-time action, enable the input prepare feature in the channel. You must enable the feature while the channel is idle. See the section called "Enabling the feature".
2. Plan the input switches and input prepares for the channel. See the section called "Planning start".
3. If the associated input switch includes input clipping, see the section called "Input clipping".

   If the associated input switch is an input failover pair, see the section called "Dynamic inputs".
4. Create the actions in the schedule. Typically, you create some prepare actions and switch actions before you start the channel for the first time. Then you add more actions over time. You add fixed switch actions, and follow switch actions. You add prepare actions as soon as you know that you will have an immediate switch some time in the future. Typically, you add all these actions while the channel is running, but you can also add them when the channel is idle.

   For detailed information on adding an input prepare action to the schedule, see *Setup: Creating a schedule*.

Topics

- Enabling and disabling the input prepare feature
- Planning the start type for an input prepare

- Input prepare and dynamic inputs
- Input prepare with clipping
- Input prepare and automatic input failover

Enabling and disabling the input prepare feature

Before you add input prepare actions to the schedule, you must enable the feature.

To enable the feature

- On the **Create channel** page, in **General settings**, in the **Feature activations** section, set **Input prepare schedule actions** to **Enabled**.

To disable the feature

You can disable the input prepare feature.

Typically, the only reason to disable input prepare is because you must attach an RTMP pull input to the channel.

1. Stop the channel.
2. Delete all active and future input prepare actions from the schedule. You don't need to delete stale input prepare actions from the schedule.
3. On the **Create channel** page, in **General settings**, in the **Feature activation** section, set **Input prepare schedule actions** to **Disabled**.
4. Attach the RTMP pull input in the usual way.

Planning the start type for an input prepare

Before you add an input prepare action to the schedule, decide on the start type for the action.

Topics

- Types of starts for input prepares
- Guidelines for choosing the start type

Types of starts for input prepares

There are three start types for input prepare actions. These start types are the same as the start types for input switches.

- Fixed – the input prepare starts at a specific time.
- Immediate – the input prepare starts as soon as you add the action to the schedule.
- Follow – the input prepare follows a specific input switch—the *reference input switch*. It can have a start or an end *follow point*—it can follow the start of the reference input or the end of the reference input.

With the follow start type, the following rules apply:

- You can't use the console to create a follow input prepare with a follow point set to *start*. The start option is not shown on the console. Only the end option is shown.
- MediaLive starts preparing the input *after* the reference input is active. Therefore:
  - For a follow-start prepare (which you can create only using the CLI), you must add the prepare action before the reference input has started in the channel.

    If the reference switch is an immediate switch, you must include the switch action and the prepare action in the same batch update command.

    If the reference switch is a fixed or follow switch, you can add the switch action in one batch update command, and the prepare action in a later batch update command.
  - For a follow-end prepare, you must add the prepare action before the reference input has ended (before ingest has ended).
- You can't create two follow prepare actions that both follow the same reference switch and the same follow point. Therefore:
  - You *cannot* create action 2 and action 4 to both follow the start of action 1.
  - But you can create action 2 to follow the *start* of action 1, and action 4 to follow the *end* of action 1.

Guidelines for choosing the start type

Following are some guidelines for deciding which start type to use with an input prepare, depending on the scenario.

Keep in mind that you can only prepare one switch at a time. When a prepare action starts, MediaLive starts preparing the input, and automatically stops any other active prepare input action.

Therefore, the guiding principle is to make sure that you don't start preparing input X and accidentally *stop* preparing input Y, if input Y needs to be prepared before input X.

Topics

Scenario A

You are flipping between two inputs. The switch start is always undetermined, so that each switch is an immediate switch. There might be more switches interspersed among these switches, but they don't need preparing.

```
Switch to input A (immediate)
Switch to input B (immediate)
Switch to input A (immediate)
Switch to input B (immediate)
```

The easiest plan is to start preparing B after each switch to A, and to start preparing A after each switch to B. You could set up each prepare input action with any of these start types:

- Fixed. The start time for prepare B is some time after the start time for switch A.
- Immediate. Recommended. You could add the prepare B action at the same time as the immediate A switch, or shortly afterward.
- Follow (start). You should add the prepare B action and the immediate A switch in the same batch update command. The reference action for the prepare B action is input A.
- Follow (end). You can add the prepare B action at any time after switch A has been added to the schedule. The reference action for the prepare B action is input A.

For example:

```
Switch to input A (immediate)
Prepare input B (immediate)
Switch to input B (immediate)
Prepare input A (immediate)
Switch to input A (immediate)
Prepare input B (immediate)
Switch to input B (immediate)
```

Scenario B

There is an immediate switch to A, then there are several fixed or follow switches. You anticipate that the next immediate switch will be to A again.

```
Switch to input A (immediate)
Switch to input C (fixed or follow)
Switch to input D (fixed or follow)
Switch to input A (immediate)
```

After the switch to A, MediaLive continues preparing A. Therefore, there is no need to prepare it again. Furthermore, if input A is the only input that ever has an immediate switch, you can prepare A once, before the first time that you switch to it. You don't need to prepare it again.

Scenario C

There is an immediate switch to A, then there are several fixed or follow switches. You anticipate that the next immediate switch will be to B.

```
Switch to input A (immediate)
Switch to input C (fixed or follow)
Switch to input D (fixed or follow)
Switch to input B (immediate)
```

You know that the next immediate switch will be to input B, so you can start preparing it anytime after the switch to input A. You could set up each prepare input action with any of these start types:

- Fixed. The start time for prepare B is at least 10 seconds before the start of switch B.
- Immediate. Recommended. You could add the prepare B action at the same time as the immediate A switch, or shortly afterward.

- Follow (start). Not recommended. You could, for example, set up the prepare B action to follow the start of switch C or the start of switch D.
- Follow (end). Not recommended. You could, for example, set up the prepare B action to follow the end of switch A or the end of switch C. Don't set it up to follow the end of switch D.

For example:

```
Switch to input A (immediate)
Prepare input B (immediate)
Switch to input C (fixed or follow)
Switch to input D (fixed or follow)
Switch to input B (immediate)
```

Scenario D

There is an immediate switch to input B, then there are several fixed or follow switches. You anticipate that there will be another immediate switch but initially you don't know if it will be to input B or input E.

```
Switch to input A (immediate)
Switch to input C (fixed or follow)
Switch to input D (fixed or follow)
Switch to input B or E (immediate)
```

When you know which input you will switch to, you can start preparing it. When you want to prepare it, the current input could be A, C, or D. You could set up each prepare input action with any of these start types:

- Fixed. The start time for prepare B (or E) is at least 10 seconds before the start of switch B (or E).
- Immediate. Recommended. Add the prepare input action as soon as you know whether the switch will be to B or E.
- Follow (start). Not recommended. You could, for example, set up the prepare B (or E) action to follow the start of switch C or the start of switch D.
- Follow (end). Not recommended. You could, for example, set up the prepare B (or E) action to follow the end of switch A or the end of switch C. Don't set it up to follow the end of switch D.

For example:

```
Switch to input A (immediate)
Switch to input C (fixed or follow)
Switch to input D (fixed or follow)
Prepare input E (immediate)
Switch to input E (immediate)
```

Input prepare and dynamic inputs

You can prepare for an input switch when the associated input is a dynamic input. A dynamic input has a variable in its path. Each time that you add the input to the schedule, you specify a *replacement string* to replace the variable with a file.

When you set up the prepare input action, you must specify this replacement string. The string must exactly match the replacement string in the switch action. If the strings are not identical, MediaLive won't prepare the input in advance.

You might use this dynamic input more than once in the channel, and the replacement string might be different in each instance. Make sure that you change the string in each prepare action.

Input prepare with clipping

You can prepare for an input switch when the associated input is a file input that includes input clipping.

When you set up the prepare input action, you must specify the start and end for the clip. The values that you enter must exactly match the start and end in the switch action. If the values are not identical, MediaLive won't prepare the input in advance.

You might use this file input more than once in the channel, and the start and end might be different in each instance. Make sure that you change the start and end in each prepare action.

Input prepare and automatic input failover

Your channel might include some inputs that are set up as automatic input failover pairs.

When you set up the prepare input action for an input that is a failover pair, make sure that you specify the primary input as the associated input (in the **Input attachment** field on the **Create schedule action** page). If you specify the secondary input, MediaLive won't prepare the inputs in advance.

When MediaLive performs the prepare action, it prepares both inputs. This means that a later input switch action can be to either of the inputs in the failover pair.

Here is a scenario that illustrates some of the key behavior:

1. You prepare input A by specifying the primary input. The prepare starts.
2. You switch to input A by specifying the primary input. The channel switches to input A.
3. You then prepare input B. The prepare starts.
4. You notice that input A is degrading, so you switch to the secondary input. You don't have to prepare input A. Even though you have started to prepare input B, the secondary input for input A is still being prepared, as part of the automatic input failover process. Therefore, the switch proceeds seamlessly.
5. You switch away from input A.
6. You prepare input A again, because you are going to switch to it later. You specify the primary input. The prepare starts.
7. You switch to input A. But you then switch to the secondary input, because the primary input is still degraded. You can switch to the secondary input because, even though you specified the primary input in the prepare action, MediaLive always prepares both inputs.

How input prepare actions behave at runtime

All prepare actions that you add to the schedule sit in the schedule until the start time. At the start time (which can be fixed, immediate, or following an input switch), MediaLive stops any input prepare that is currently active, and starts the new input prepare.

Eventually, MediaLive switches to the associated input. At this point, MediaLive doesn't stop preparing the input. The input prepare continues either indefinitely or until another input prepare starts. This perpetual prepare characteristic can be useful. For an example, see scenario B.

If a channel fails, MediaLive automatically restarts the channel. If the schedule indicates that there is an upcoming immediate switch action, and the schedule also contains a prepare action for that input, then MediaLive starts preparing the input again. You don't need to take any steps.

Modifying input prepare actions

For information on modifying an input prepare action, see the section called “Modifying actions”.

Deleting and stopping input prepare actions

You can delete input prepare actions from the schedule. There are different rules for deleting actions depending on the current state of the channel. The channel can be running, idle, or recovering. The channel is idle if you manually stopped it. The channel is recovering if it failed and MediaLive is automatically restarting it.

For detailed information on deleting an action, see the section called “Deleting actions”.

Deleting actions while the channel is running

When the channel is running, you can't delete the most recent input prepare action that is in the past. This rule exists because the associated input switch might be in the future. When MediaLive automatically restarts the channel, it must also restart the input prepare, to ensure that the input for the immediate input switch will be prepared.

Deleting actions while the channel is idle

When the channel is idle, you can delete any input prepare action.

Stopping an input prepare

To stop an active input prepare, add an immediate input prepare with no input specified.

For detailed information on adding an action, see the section called “Creating actions”.

Input switching in AWS Elemental MediaLive

You can set up an AWS Elemental MediaLive channel to ingest multiple sequential inputs, rather than setting it up to ingest only one input. You set up this *multiple-input channel* by attaching more than one input to the channel, and then adding actions in the channel's schedule that specify when to switch from one input to another.

Topics

About multiple-input channels and input switching

You set up input switching in a channel in order to ingest the inputs in a multiple-input channel.

Topics

Multiple-input channels and the schedule

Input switching works as follows: You create a channel that contains more than one input attachment. After the channel is created, you go into the schedule for that channel and add input switches, to create rules for moving from one input attachment to another. When you start the channel, the channel will automatically switch inputs according to the schedule.

To work successfully with multiple-input channels, remember the following.

The schedule exists inside the channel

The schedule does not exist separately from the channel. On the console, you find the schedule in the details page for an existing channel.

There is no implicit switching

With a multiple-input channel, you must add input switches to the schedule to instruct the channel to switch. A channel that contains more than one input attachment won't switch to the next input attachment in the list of input attachments unless the schedule specifies to do so.

There is no "main" input

With a multiple-input channel, you must think of the input attachments as a *pool* of inputs all with equal status. There isn't one input that is the main input, that the channel returns to when it has nothing else to ingest.

Typical use cases

Scheduled input switching supports the following use cases.

Use case 1: One live feed and one file input alternating

You have a channel to process a live (streaming) feed from a specific source, perhaps for a sports tournament. Periodically (perhaps between individual sports events), the live feed should be replaced by file content (perhaps a filler such as a video of ocean waves). After a few minutes, the same live feed should be resumed.

You set up the channel with one live input and one file input. The first input is the live input.

Before you start the channel, you create a schedule that consists of actions to switch to the live input at the top of each hour—at 10:00 AM, 11:00 AM, and so on.

You then start the channel. As soon as each sports event has finished, you modify the schedule "on the spot" to switch to the video filler. The live feed continues for a few moments (perhaps showing the sports crowd or the players leaving the stadium), and then the channel switches to the filler video. At the top of each hour, the channel switches to the live feed.

Use case 2: One live feed and file inputs, and the channel starts with a file input

You have the same requirements as for use case 1, except that you want to start the channel with a file clip, perhaps from the opening of the sports event. At the top of the first hour, you want to show the video filler. But at the top of the second and succeeding hours, you want to show highlights from earlier in the day.

You set up the channel with one live event (a live input) and several file inputs: one for the opening, one for the video filler, and several for the highlights. The first input is the file input for the opening event.

Before you start the channel, you create a schedule that contains one action to switch to the live input as soon as the file input has finished.

You then start the channel. As time goes on, you modify the schedule to add more actions, as for use case 1, to switch back and forth between the live input and the file inputs.

Use case 3: Two live feeds

You have a channel to process live feed from two different sources. You want to insert ad content into the channel, as required. You want to insert this ad content using MediaLive. (You don't want to insert SCTE-35 messages that a downstream system will read in order to replace the avails with ad content.)

The live feeds might be the venue feed and the in-studio feed for the same sports event. You want to switch from one live feed to the other. You want to time the switches "on the spot" instead of according to a strict clock schedule. Occasionally, you want to switch from one live feed to an ad. When the ad is finished, you might want to return to one of the live feeds.

You set up the channel with two live inputs and several file inputs (one file for each ad).

Before you start the channel, you create a schedule that contains the first action in the schedule. That action is to switch to the first input, input A, that you want to the channel to ingest. You set the start time for input A to a time that is at least one minute earlier than the time that you start the schedule. You then start the channel. MediaLive immediately reads the schedule and switches to the input that is supposed to be the current action, which is input A. When appropriate, you modify the schedule on the spot to add actions to queue up one or more switches.

Use case 4: VOD-to-live

You have a channel to process only MP4 file inputs, or mostly MP4 file inputs, on a 24/7 basis.

You set up the channel with a series of file inputs to run one after another. Each file is encoded from start to finish, and then the next file starts. Sometimes, you want to clip a file and play only part of that file.

You want this channel to run without stopping, until the next scheduled maintenance period, which might be in several weeks.

To overcome the limit of 20 inputs per channel, you take advantage of the *dynamic input* feature. You create some file inputs with a variable in the place of all or part of the path and file name. You set up the schedule to use this dynamic input over and over again, each time with a different file name slotted into the variable. You can set up several dynamic inputs.

Fixed, immediate, and follow switches

You can categorize input switches according to the start types for the switch.

- Fixed – A fixed input switch starts at a specific time.

  Fixed switches use UTC time. They don't use the timecode of the input.
- Immediate – An immediate input switch starts as soon as possible. This type of switch is more like a fixed switch than a follow switch because it interrupts the current input. The advantage of this switch over a fixed switch is that you don't have to calculate any buffer in the start time.

- Follow – A follow input switch starts when the previous input has ended (when MediaLive has reached the end of the file).

This start type is a property of the switch, not a property of the input itself. Therefore, in the schedule you can switch to a specific input with a fixed switch, and then later switch to the same input with a follow switch.

Types of switches and types of inputs

The combination of types of switches and types of inputs (file and live) means that there are these types of switches:

- A file input with a fixed start. The previous input can be a file or live input. At the specified start time, MediaLive stops ingesting the previous input and switches to the new input.
- A file input with an immediate start. The previous input can be a file or a live input. As soon as possible after you enter this switch in the schedule, MediaLive stops ingesting the previous input and switches to the new input.
- A file input that follows the previous input. The previous input must be a file input. It can't be a live input because a live input doesn't have an end, so the switch would never occur.
- A live input with a fixed start. The previous input can be a file or live input. At the specified start time, MediaLive stops ingesting the previous input and switches to the new input.
- A live input with an immediate start. The previous input can be a file or a live input. As soon as possible after you enter this switch in the schedule, MediaLive stops ingesting the previous input and switches to the new input.
- A live input that follows the previous input. The previous input must be a file input. It can't be a live input because a live input doesn't have an end, so the switch would never occur.

The following table summarizes the inputs and start types.

Current Input	Next Input	Possible Start Type
File	File	Fixed or Immediate
File	File	Follow
File	Live	Fixed or Immediate

Current Input	Next Input	Possible Start Type
File	Live	Follow
Live	File	Fixed or Immediate
Live	Live	Fixed or Immediate

Follow chains

A series of follow input switches is called a *follow chain*. When each input ends, MediaLive automatically starts ingesting the next input. Here is a diagram of a follow chain:

```
Input A    Fixed or Immediate   File
  Input B  Follow               File
  Input C  Follow               File
  Input D  Follow               File or Live
Input E    Fixed or Immediate   File or Live
```

The follow chain starts with the *reference action*—the input above the first follow. It ends with the last follow input. In the preceding example, the chain starts with the reference action input A and ends with input D. Inputs A, B, and C must be files because they must have a defined ending so that the next input can successfully follow. Input E breaks the chain because it is fixed or immediate.

Static inputs and dynamic inputs

A file input in a multiple-input channel can be set up as a *static input* or a *dynamic input* (A live input is always a static input). The following examples use MP4 files, but note that both MP4 and Transport Stream (TS) files are supported.

- With a static input, the source content of the input is always the same file. For example, `s3ssl://DOC-EXAMPLE-BUCKET/my-movie.mp4`.
- With a dynamic input, all or part of the source content of the input is a variable. For example, `s3ssl://DOC-EXAMPLE-BUCKET/movies/$urlPath$`. Each time you set up to switch to this input, you replace the variable with a different file. For example, `s3ssl://DOC-EXAMPLE-BUCKET/movies/my-movie.mp4` in one input switch and `s3ssl://DOC-EXAMPLE-BUCKET/movies/mlaw.mp4` in another input switch.

Using dynamic inputs lets you increase the number of video sources that you can use in the channel, while still observing the limit on the number of inputs that you can attach to the channel.

To set up a dynamic input, you create the input with a variable as all or part of the URL of the file. Then, in the schedule, when you create an input switch that uses that input, you replace the variable with a real filename.

The procedure for setting up for input switching, later in this section, provides detailed information about deciding whether you should set up some inputs as dynamic inputs.

Input Prepare

The schedule includes an input prepare action that is a helper action for input switches.

For more information about input prepare, see the section called “Input prepare”.

Rules and limits for input switches

This section describes the rules and limits that apply to input switches.

Rules for types of inputs

There is flexibility in the number and types of inputs that you can set up for input switching. For example:

- You can have both HLS live inputs and MediaConnect inputs attached to one channel.
- You can have both RTMP push inputs used for a source from the public internet and an RTMP VPC push input.

But there are also some restrictions:

- The number of push inputs and pull inputs that you can attach to a channel.
- The number of inputs of a specific input type. For example, the number of CDI inputs you can attach to a channel.
- Use of VOD assets.
- Use of inputs in different Availability Zones.
- Use of dynamic inputs in an input switching workflow.

For detailed information about these rules, see *Feature rules and limits*.

First switch must be static

The first switch in the channel must be for a static input. It can't be a dynamic input.

No limits to the number of input switches

The schedule for the channel can contain any number of scheduled input switching actions.

You can switch to a specific input as many times as you want.

Reusing a file input

If you switch away from a static file input and then switch back to it, the channel ingests the file from the start of the file or start of the file clip (if you clipped the file). This rule applies even if you switch away from the file input before the end of the file.

This rule also applies if you switch away from a dynamic file input and then switch back to it without changing the value of the variable portion of the URL. The channel always ingests from the start.

Setting up for input switching

When you plan for a channel that includes multiple inputs, there are special requirements that you must consider.

This section assumes that you are familiar with the general procedures for designing a channel, as described in *Setup: Planning the channel* and for creating a channel, as described in*Setup: Creating a channel* .

Topics

Step 1: Plan the outputs

Plan the output side of the channel in the normal way:

- Identify all the output groups.
- Identify the types of outputs in each output group.
- Identify the video, audio, and captions encodes for each output.

For more information, see *Setup: Preparing upstream and downstream*.

After you have completed this step, you have a list of output group types, and a list of the number of video, audio, and captions outputs in each output group.

Step 2: Assess the sources

When planning a multiple-input channel, you must identify all the sources that you need. You must then assess the audio and captions in each source to ensure that the source is suitable for an input-switching scenario.

Result of this Step

After this step, you have a set of sources that you can successfully set up as inputs and attach to the channel in order to implement input switching in the channel. You have categorized these sources by their type: live sources or file sources.

Topics

- Identify the sources
- Assess the video in the sources
- Assess the audio in the sources
- Assess the captions in the sources

Identify the sources

1. Identify all the sources that you will need through the lifetime of the channel or at least until the next planned maintenance period.
2. Note which sources are push inputs and which are pull inputs. Make sure that you don't exceed the limits.

3. Note which sources are live sources and which are file sources. For information on whether a source is a live or file (VOD) source, see the section called “Input types”.

Assess the video in the sources

There are no special requirements for the video when planning a multiple-input channel. Assuming that AWS Elemental MediaLive supports the video codec that is in a source, you can use that source as an input for the channel.

There is no requirement for the sources to have matching video codecs.

Assess the audio in the sources

MediaLive provides flexibility in extracting audio from sources in a multiple-input channel. It also has some special requirements for the audio in these sources.

To assess the audio in the sources

1. Read the information lower down about flexibility to get a sense of how MediaLive supports a wide variety of audio sources.
2. Then read each of the requirements for information on specific constraints in the audio sources. Make sure that the audio in each source meets these requirements.
3. If you reject a source, you might want to contact the upstream system to determine if it could provide a more suitable version of the source content.

Flexibility in using audio

When assessing the audio, note the following rules. These rules provide flexibility in extracting audio, and therefore allow you to use a variety of sources:

- Different languages in a source can use different codecs. For example, in your sources English might be in AAC while Spanish is in MPEG-2.

- The method of identifying an audio language in the source doesn't have to be the same in all the sources in the multiple-input channel.

  For example, in source 1 you can identify the languages by PID. In source 2, you can identify by language code.

First requirement: each language must have the same coding mode in all sources

Each output language must be present in every source, and the coding mode must be the same in all sources.

For example, assume that the channel contains an Archive output group that contains one audio encode for English 2.0 and one audio encode for French 2.0:

- Assume that you have a source that contains AAC 2.0 audio in English and Dolby Digital 5.1 in French.
- Assume that you have a second source that contains AAC 2.0 audio in English and AAC 5.1 audio in French.

  For English, this source contains audio with the same codec and coding mode as the first source. For French, it contains the same coding mode as the first source but a different codec.

  This source is acceptable. The fact that in a comparison of source 1 and source 2, the codecs are different for French isn't relevant. The requirement is that the *coding modes* are the same.
- Assume that you have a third source that contains AAC 2.0 audio in English and AAC 2.0 audio in French.

  This source is *not* acceptable because for French, the audio has a different coding mode from the first source.

Second requirement: each language must provide the highest coding mode required

For each language, every source must include audio that can produce all the highest coding mode among all the outputs in the channel.

For example, assume that the channel contains an Archive output group that contains one audio encode for Spanish AAC 2.0. The channel also contains one HLS output group that contains one audio encode for Spanish Dolby Digital 5.1:

- Assume that you have an source that contains Dolby Digital 5.1 audio in Spanish.

  This source contains audio that can produce all the desired output audio encodes for Spanish. You must set up the Archive output to remix the audio down to 2.0. You don't need to set up the HLS output to remix the audio.
- Assume that you have a second source that contains AAC 2.0 in Spanish.

This source is *not* acceptable. This source can't produce Spanish Dolby Digital 5.1 for the HLS output.

Third requirement: mp4 sources should not contain variations of the same language

An MP4 file that contains multiple variations of a language might produce undesirable output audio. For best results, the file should contain only one version of a language:

- For example, assume that one MP4 source contains AAC 5.1 audio in English. The channel output requires one audio encode for English 2.0. Therefore, in the output you set up the audio encode to down mix from 5.1 to 2.0.
- Assume that you have a second source that contains AAC 2.0 in English in track 2, and Dolby Digital 5.1 audio in English in track 3.

  MediaLive extracts audio from MP4 files by language code and it extracts from the first track that contains that language. In this example, it extracts track 2, which contains AAC 2.0. It ignores track 3. On the output side, MediaLive will try to remix this source, resulting in audio that has poor quality.

Fourth requirement: all sources must contain dolby if producing passthrough encode

If one of the outputs includes an encode that is set up with the Passthrough codec, then all the sources must include Dolby Digital, Dolby Digital Plus, or Dolby Atmos in all the required language or languages.

If any single source doesn't include one of these codecs, you can't use it in the multiple-input channel.

The Passthrough option for a codec allows for the ability to ingest audio that is in Dolby Digital, Dolby Digital Plus, or Dolby Atmos and in any coding mode, and pass it through without transcoding it.

Assess the captions in the sources

There are special requirements for the captions in sources for a multiple-input channel.

To assess the captions in the sources

1. Read each of the requirements that follow for information on specific constraints in the captions sources. Make sure that the captions in each source meets these requirements.
2. If you reject a source, you might want to contact the upstream system to determine if it could provide a more suitable version of the source content.

First requirement: a source must contain all required captions languages and formats

With a multiple-input channel, for every output there must be a captions asset in the source that can produce the captions in that output. If a source doesn't have all the source captions to produce all the output captions, it can't be used as a source in a multiple-input channel.

For example, assume that the channel contains an Archive output group that contains one output with one captions encode for embedded captions in English, French, Spanish, and German. The channel also contains one HLS output group that contains four captions outputs, one each for English, French, Spanish, and German Web VTT captions.

Every source must include a captions source that can produce both embedded and Web VTT captions. The source can contain one captions source that can produce both output types, or the source can contain two captions sources:

- Assume that you have a source that contains embedded captions in the four languages.

  This source is acceptable because embedded captions can produce embedded captions in the output and Web VTT captions in the output.
- Assume that you have a source that contains DVB Sub in the four languages.

  This source is *not* acceptable because DVB Sub captions can't produce embedded captions in the output.
- Assume that you have a source that contains embedded captions in English, French, German, and Bulgarian.

  This source is *not* acceptable because one of the languages is Bulgarian instead of Spanish.
- Assume that you have a source that contains embedded captions in English and French.

  This source is *not* acceptable because it is missing two of the output languages.

Second requirement: for embedded passthrough all sources must contain languages in the same order

When there is at least one output that has embedded captions and there are at least two sources that have embedded captions, the languages must be in the same order in those sources.

Passthrough means that an output requires embedded captions encodes in one or more languages, and a source contains embedded captions (typically in four languages). For example, the output requires English and Spanish embedded captions. A source contains embedded captions in English and Spanish, and possibly in two other languages.

If two sources have the embedded captions languages in a different order, you can't use both the sources in the multiple-input channel. You must use only one of the sources.

Look again at the example from the preceding requirement:

- Assume that you have a source that contains embedded captions with the languages in the four channels in this order: English, French, Spanish, and German.

  Assume that you have a second source that contains embedded captions with the languages in a different order: French, Spanish, German, and English.

  Only one of these sources is acceptable.

When this scenario applies to your channel, you should decide which sources to keep and which ones to reject. One rule you could follow is the following:

- Compare the order of the captions languages in those sources.
- Identify the order of the most important source, or identify the order that most sources follow.
- Accept only the sources that follow this order. Reject the other sources.

> **Note**
>
> This requirement applies only to embedded passthrough.
> If the channel doesn't contain any outputs that contain embedded captions, then you can use any source that contains embedded captions because the order of the languages in the sources isn't relevant. The embedded captions aren't passed through. They are converted to another format, such as DVB-Sub.

Step 3: Organize sources into static and dynamic inputs

This section is a supplement to the information in the section called "Inputs". It provides information that applies to inputs used in a multiple-input channel.

After you follow step 2 to assess the sources, you end up with a set of sources that are suitable for your multiple-input channel. You must now organize these sources into three types of MediaLive inputs: static live inputs, static file inputs, and dynamic file inputs.

Result of this step

After this step, you have a list of the following:

- Sources that you will set up as static live inputs. Each source becomes one input (and one input attachment).
- Sources that you will set up as static file inputs. Each source becomes one input (and one input attachment).
- Sources that you will set up as dynamic files inputs. Several sources become one input (and one input attachment).

Identify the live sources

Make a note of the sources that are live sources. Each of these sources becomes a static live input.

Identify and organize file sources

You must assess your files sources and determine if you should implement some sources as dynamic inputs, rather than as static inputs.

A static input is always associated with the same source. A dynamic input can be associated with a different source each time that you attach it to the channel. It is therefore more flexible and can help you work with the limit on the number of inputs attached to a channel. For general information about dynamic inputs, see the section called "Dynamic inputs".

To organize the sources

1. Organize the file sources into sets, where the sources in each set are all stored in the same source location with the same access credentials, such as the same bucket in Amazon S3.

For example, you might have a set of file sources in the bucket called "prerolls," and another set in the bucket called "filler". Each bucket has different access credentials, so each one is its own set.

2. Read this step if you have inputs with embedded captions that you are converting (instead of passing through). If you don't have inputs with embedded captions, or if you do have inputs with embedded captions but they are always passed through to the output, then skip this step.

   - Within each set, identify the file sources that contain embedded captions. Determine if there is at least one output that is converting these captions rather than passing them through.
   - In each file source that contains embedded captions, identify the order of the languages.
   - Where necessary, subdivide the set according to language order.

     For example, you might have one set of file sources in an Amazon S3 bucket where the languages are in the order English, French, Spanish, and German. You might have another set in the same bucket where the order is French, Spanish, German, and English. Divide this set into two sets.

3. Make a list of the sets that you identified. For example, you might have these sets:

   - File sources from the Amazon S3 "preroll" bucket with embedded captions in the order English, French, Spanish, and German
   - File sources from the Amazon S3 "filler" bucket with embedded captions in the order French, Spanish, German, and English
   - File sources from the Amazon S3 "filler" bucket with embedded captions in a different order, such as English, French, Spanish, and German

4. Decide whether each set of file sources becomes a static file input or a dynamic file input. Follow these rules:

   - Any set that contains more than one file source becomes one dynamic input.
   - Any set that contains only one file source can become a static input. However, if you think you might later use other file sources from that location (for example, from that Amazon S3 bucket), you might want to treat the set as a dynamic input, in order to not exceed the limit for file inputs.

Step 4: Design the selectors for each input

After you follow step 3 to organize sources into different inputs and input types (static and dynamic), you must identify the content to extract from each input.

Result of This Step

After this step you have:

- Names for all the inputs
- A list of video, audio, and captions selectors for each input

Topics

Plan the input and input attachment names

You should plan the names for the input and the input attachment. Here are some tips:

- Use the same name for the input and input attachments.
- Include an indicator of whether the entity is static or dynamic.
- For a static input, include either the name of the video source or a description of the video source.
- For a dynamic input, include an indicator of its characteristics, which you determined in step 2. Doing so ensures that you do not attach an unsuitable video source when you specify the URI in the input switch action.

For example, for a static input:

- `static-filler`
- `static-live-studio-feed`

For example, for a dynamic input:

- `dynamic-s3-preroll-bucket-embedded-EN-FR-ES-DE`
- `dynamic-s3-preroll-bucket-embedded-FR-ES-DE-EN`

Plan the video selectors

You can extract only one video from each input. If a given input contains more than one video, then create a video selector to extract that specific video. If a given input contains only one video, there is no need to create a video selector. AWS Elemental MediaLive automatically finds and extracts that video. On the output side, MediaLive automatically uses that one video asset.

Plan the audio selectors

There are several rules you must follow when planning the audio selectors. When you set up the audio selectors for an input, you specify the language to extract but you don't specify the format of the audio in that input. AWS Elemental MediaLive extracts that input so it can be included in the output. The output expects to be able to find the specific extracted language.

Rule 1: Plan the same number of selectors in every input

The selectors in each input must extract sufficient assets to produce every output audio encode. In addition, every input must have the same number of selectors.

For example, assume you have an output that requires AAC 2.0 audio in English and French. You have a second output that requires Dolby 5.1 audio in English and French. You have a third output that requires Dolby 5.1 audio in French, Spanish, and Portuguese:

- If the first input contains Dolby Digital 5.1 in the four languages, you must create four selectors —one for each language. The audio extracted by these four selectors can produce all the languages. It can produce Dolby Digital 5.1 for the first output, and it can produce AAC 2.0 for the second because you can set up that output for remixing.

  Although the channel has seven output audio encodes, you don't need seven selectors.
- If the second input contains Dolby Digital 5.1 in French (but no other language), and also contains AAC 2.0 in English, Spanish, and Portuguese (but not in French), you create four selectors. The selector for French will find that audio only in the Dolby Digital 5.1. The selectors for the other languages will find those audio assets only in the AAC 2.0.
- If the third input contains Dolby Digital 5.1 in the four languages, and also contains AAC 2.0 in the four languages, you still create only four selectors.

Although you might think to create selectors to extract the AAC 2.0 audio for French and English just for this input, you mustn't do this because the first input doesn't have these selectors. Remember that every input must have the same number of selectors.

Rule 2: Plan a separate selector for Dolby Digital Plus 7.1

If the channel includes at least one output with Dolby Digital Plus 7.1, create one selector in every input for that audio asset. On the output side, in every audio encode for Dolby Digital Plus 7.1, you will map the audio encode to that selector.

After you have identified all the selectors for all the inputs, you might end up with a list like this:

- Selector for English
- Selector for French
- Selector for Spanish
- Selector for Portuguese
- Selector for EAC3 passthrough (EAC3 is another name for Dolby Digital Plus)

Each of these selectors applies to all inputs, regardless of the audio format in that input.

Rule 3: Plan the same selector names in every input

Every selector for a specific language must have the same name across all the inputs. This rule exists because each output references the selectors only once. The output doesn't reference the selector once for each different input.

We recommend that you give the selectors names that include the language. Don't include the format unless you create a selector for Dolby Digital Plus 7.1.

Planning the captions selectors

When you set up the captions selectors for an input, you specify both the format and the language to extract from the input. Each input has the number of selectors that is appropriate to the captions formats in that input. Therefore, each input might contain a different number of selectors. The method for extracting captions is different from the method for extracting audio.

Rule 1: Plan the number of selectors for an input that is appropriate to the input and output

In each input, you must create the number of selectors that is appropriate to the input format and output format:

- For example, if you want to extract embedded in order to pass through the captions, you create one selector.
- If you want to extract embedded in order to convert them to TTML, you create one selector for each language.

After you have identified all the selectors for all the inputs, you might end up with a list like this:

- Selector for embedded passthrough – applies to input 1, input 3, and input 4
- Selector for embedded, English – applies to input 1, input 3, and input 4
- Selector for embedded, French – applies to input 1, input 3, and input 4
- Selector for DVB Sub, English – applies to input 2
- Selector for DVB Sub, French – applies to input 2
- Selector for Teletext passthrough – applies to all inputs

Note that inputs 1, 3, and 4 each contain four selectors. Input 2 contains three selectors.

Rule 2: Plan the same selector names in every input

Every unique selector must have the same selector name across all the inputs. This rule exists because each output references the selectors only once. The output doesn't reference the selector once for each different input where the selector exists.

We recommend that you give each selector a name that includes the language and the source format. Descriptive names help you to choose the correct selector on the output side.

Step 5: Plan the input switches in the schedule

After you design the selectors for each input (step 4), you must plan the order that you want MediaLive to follow when it ingests these inputs.

Result of This Step

By following this step, you have identified one input as being the first that you will add to the channel.

You have also identified an ordered list of input switches. You have the following for each switch:

- An action name for the switch.
- The name of the input attachment associated with the switch.
- The switch input identified as either static or dynamic.
- The type of switch—fixed, follow, or immediate.

Topics

Plan the action names

You should plan the names for the input switch action. Action names must be unique in the schedule for each channel.

For a static input, you might want to name the actions so that they indicate which input applies. For example, for each switch to the input named static-live-studio-feed:

- `static-live-studio-feed-action-1`
- `static-live-studio-feed-action-2`
- `static-live-studio-feed-action-3`

For the input switch action for a dynamic input, you might use the input name (or part of the name) plus the URL (or part of the URL) of the file. For example:

- `dyn-preroll-EN-FR-ES-DE-ad-ward-cars-1`
- `dyn-preroll-EN-FR-ES-DE-ad-zel-cafe`
- `dyn-preroll-EN-FR-ES-DE-ad-ward-cars-2`

Plan the order of input switches

We recommend that you plan the order of the input switches before you create the actions in the schedule using the console or the CLI.

To plan the order of input switches

1. In the first position, put the input attachment that you want MediaLive to ingest first. Make a note that this input will be an immediate switch in the schedule.
2. Make a list of switches and the input attachment to use for each switch. Decide on the start type for each switch—fixed, immediate, or follow. For more information, see the section called "Fixed, immediate, and follow switches" and the section called "Rules and limits".

   You should be able to organize the fixed and follow input switches into an ordered list. You might not be able to include the immediate switches in the ordered list because you don't know their start times. See the example after this procedure.

   Note the following about switching to an input:

   - You can switch to an input attachment as many times as you want.
   - When you switch to a dynamic input, you must provide the URL that applies for that usage of the dynamic input. In the list that you make, specify the URL for each usage.
3. Read the information later in this section about handling the transition between switches. For each input attachment in your list, make a note of how to handle the transition.

About Models for the Schedule

There are two models for setting up input switches in the schedule:

- In the recommended model, you use only the schedule to control the ingest of all inputs. With this model, the order of the input attachments in the channel isn't relevant. You set up the schedule so that the first input switch is an immediate switch to the input that you want to ingest first. As soon as the channel starts and before the channel starts to ingest, the channel performs that immediate switch.

  The steps earlier in this section show how to design the schedule for this model.
- In the other model, the first input attachment is the first input that MediaLive ingests. You set up the schedule to perform its input switch only after that first ingest.

We don't recommend this model because you must look at the order of input attachments and at the schedule. With the first model, you monitor the order of ingest from one place—the schedule.

Example of a list of input switches

This example shows a list of planned input switches. The first input is an immediate switch to a file input. Then there are several short file inputs that are follow switches, so that the switch occurs at the end of the previous input. These inputs run one after another, but the plan is to interrupt these at any time with an immediate switch to the first live input. After that, the schedule switches back and forth between two live inputs. You don't know the exact timing for the switches, so you will set up these switches as immediate switches.

Ordered list: action name, start type, input attachment name

- startup, immediate, banner
- static-1, follow, short-clip-12
- static-2, follow, short-clip-32
- static-3, follow, short-clip-77
- static-4, follow, short-clip-18

Immediate switches to occur at any time:

- static-live-studio, immediate, live-1
- static-live-alternate, immediate, live-2

Handling the transition when the next input is fixed or immediate

When planning the schedule, you should ensure that there is no gap when switching from a file input (input A) to an input (input B) that starts at a fixed time or that starts immediately. Input B can be a file or a live input. If the current input ends before the switch start time, there is potential for a gap.

The **Source end behavior** field in each input attachment controls the gap. (This field appears in the **Input attachments** page, in the **General input settings** section of the channel.) There are two options to ensure a smooth transition in this situation:

- If you set the **Source end behavior** field for input A to **LOOP**, then when input A finishes, MediaLive goes back and ingests it again until the start time of input B occurs.
- If you set the **Source end behavior** field for input A to **CONTINUE**, then input A is ingested only once; when the input finishes, the channel follows the behavior specified in the **Input Loss Behavior** set of fields (although without the "repeat frames" logic). When the start time of input B occurs, the input loss behavior ends and the channel switches to input B.

  (To display this field, in **General input settings** for **Global configuration**, for **Input loss behavior**, choose **Input loss behavior**. More fields appear. For more information, see the section called “Input loss handling”.)

Handling the transition when the next input is follow

When planning the schedule, you should ensure that a switch from one input to a "follow input" can succeed.

A follow input (input B) won't succeed if the current input (input A) is set up to loop. When AWS Elemental MediaLive reaches the file end, it starts to ingest again from the beginning of the file.

The **Source end behavior** field in each input attachment controls looping. (This field appears in the **Input attachments** page, in the **General input settings** section of the channel.)

- Always set the **Source end behavior** for input A to **CONTINUE**. When input A finishes, the channel immediately switches to input B.

When you create the channel, it is important to set the **Source end behavior** to **CONTINUE** in every input attachment where the next planned input in the schedule will be a follow input. If you don't set up the input with **CONTINUE**, you won't be able to set up the schedule with the next input as a follow input. You will have to cancel the schedule action, modify the input attachment, and try the schedule action again.

Prepare input—reducing latency when the next input is immediate

You might have an input switch that you have identified as an immediate input switch, but you don't know when the switch will need to occur. You only know that you will be given just a few seconds advance notice. In this situation, you might want to prepare the input in advance by creating a prepare input action. For more information, see the section called “Input prepare”.

Step 6: Create the inputs and channel

After you perform the planning in steps 1 to 5, you are ready to create the inputs and create the channel.

In a multiple-input channel, all the inputs must already exist in the channel before you start the channel. You can't add an input while the channel is running. Therefore, you should identify all the inputs that you might need until the next planned maintenance period.

Topics

Create the inputs

This section is a supplement to the information in the section called "Inputs". It provides information that applies specifically to creating inputs for use in a channel that contains multiple input attachments.

Follow the steps in the section called "Creating an input"for creating a channel, with the following notes.

- Create the inputs that you identified in the previous steps in this section.
- Make sure that you set up each input as the correct type (static live, static file, or dynamic file).

  There are no special steps for creating a static live input or static file input.

  To create a dynamic input, you must enter a variable in the URL for the file source. When this variable is present, MediaLive recognizes the input as a dynamic input. For more information, see the section called "Dynamic inputs".

Identify the first input for the channel

Identify an input that you will set up as the first input in the list of input attachments for the channel:

- This input won't be the first input to ingest because you will use the schedule to switch to the first input to ingest.

- It can't be a dynamic file input. It must be either a live input or a static file input in order for the channel to start.

Create the channel

This section is a supplement to the information in *Setup: Creating a channel* . It provides information that applies specifically to creating a channel that contains multiple input attachments.

Note the following points, and then follow the steps for creating a channel as described in *Setup: Creating a channel* .

Channel and input details pane

On the **Channel and input details** pane for the channel, in the Input specificationssection, set up each option to meet or exceed the most demanding of your inputs.

Input attachments pane

On the **Input attachments** pane for the channel, set up the input attachments for the inputs that you created.

To set up each input attachment

1. Choose **Add** in the **Input attachments** pane.
2. Choose an input. Enter the name that you decided on when you planned the attachments.
3. Choose **Confirm** to display fields for general settings, for video selector fields, audio selector fields, and captions selector fields.
4. Complete these fields as appropriate.

Note the following points:

- Attach all the inputs that you identified. If you omit an input, you won't be able to attach it unless you stop the channel.

  You should have already identified the first input attachment. Make sure that you create this attachment first, so that it appears first in the channel.
- Add the remaining input attachments in any order.

- In the **General input settings** section for each input attachment, set **Source end behavior** to work correctly. For information, see the section called "Handling the transition when the next input is fixed or immediate".
- In the **General input settings** section for each input attachment, set up the following sets of fields according to the plan that you created when you planned the attachments:
  - The fields in **Video selector**
  - The fields in**Audio selectors**
  - The fields in **Caption selectors**

Output groups

On the **Output groups** pane for the channel, follow the regular procedure to create all the output groups that you identified in the section called "Step 1: Plan outputs".

Step 7: Set up the schedule with input switches

After you create the inputs and the channel (step 6), you must create actions in the schedule to set up the input switches that you want. For detailed information about creating input switch actions, see the section called "Creating actions".

Follow these guidelines when setting up the schedule:

- You should create at least some of the fixed input switches and follow input switch actions before you start the channel.
- The first input switch in a new channel should be an immediate input switch. You should create this input switch before you start the channel. Setting up in this way ensures that the order of ingest of inputs is always being controlled by the schedule.
- For other immediate switches, you might be able to add the switches to the schedule before you start the channel. Or you might be able to add them only after the channel is running. You should have an idea of which of these strategies applies to your plan.
- Plan to update the schedule regularly. Remember that you can add actions to the schedule without stopping the channel.

Deleting actions from the schedule

You can delete input switch actions from the schedule. There are different rules for deleting actions depending on the current state of the channel. The channel can be running, idle, or recovering. The

channel is idle if you manually stopped it. The channel is recovering if it failed and MediaLive is automatically restarting it.

Deleting actions while the channel is running

When the channel is running, there are restrictions on the input switch actions that you can delete. MediaLive must preserve information about the currently active input. It must preserve that information so that if the channel fails, MediaLive can recover and start ingesting on the appropriate input. Therefore, this rule applies:

- You can't delete the most recent fixed or immediate input switch. The term *most recent* means one of the following:
  - The input is the input currently being ingested. So the most recent input and the active input are the same.
  - The input is the fixed or immediate input switch that most recently ingested. The active input might be a follow input.
- You can't delete any of the actions in a follow chain that follows this most recent fixed or immediate input switch. For example, in the following diagram, assume that input A is the most recent fixed or immediate input switch. You can't delete actions B, C, or D. You can delete E, which is not part of the follow chain.

```
Input A    Fixed
  Input B  Follow
  Input C  Follow
  Input D  Follow
Input E    Immediate
```

Deleting actions while the channel is idle

You can delete an input switch action when the channel is idle, so long as the action is still in the schedule.

To delete an action that is in a follow chain, you must delete the entire follow chain, then recreate the follow chain but omitting the unwanted action. See the section called “Deleting actions”.

Deleting actions while the channel is recovering

You can delete input switch actions while the channel is recovering.

Starting and restarting a channel that has multiple inputs

After you create the channel and add actions to its schedule, you can start the channel.

Before you start the channel, make sure that the inputs attached to the channel are ready:

- Push inputs must be already pushing before you start the channel. A push input must be already pushing even if it isn't the first input in the channel.
- If the first input in the channel is a file input, it must be ready to be pulled.
- A file input that isn't the first input doesn't have to be ready to be pulled until approximately 30 seconds before the switch to the input occurs.

Topics

What happens at runtime

When you start the channel, AWS Elemental MediaLive takes a short time to get the channel ready to run.

As soon as the channel is ready, MediaLive looks at the schedule to determine if there is an input switch with an immediate switch, with a start time that is now or with a start time that is overdue:

- If it finds this action, it switches to that input and starts ingesting.
- If it doesn't find this action, it starts ingesting the first input attachment listed in the channel.

If you set up the channel and schedule as recommended, then as soon as the channel is ready, it finds an immediate switch to the first input that you want MediaLive to ingest.

Restarting a channel

If you restart a channel that has multiple inputs set up for scheduled input switching, AWS Elemental MediaLive looks at the schedule to determine which input should currently be running. MediaLive then behaves as follows:

- If that input is a live input, then MediaLive starts ingesting that input at the current frame.
- If that input is a file input set to start at a fixed time or immediately, then MediaLive starts ingesting that input at the start of the file or of the file clip (if you clipped the input). It doesn't adjust for the difference between the scheduled time and the current time. For example, assume that it is now 13:10:00 UTC. The schedule specifies to switch to input X at 13:00:00. MediaLive starts ingesting the file from the start, not from 10 minutes into the file.
- If the current input is ambiguous because there is a chain of follow inputs, then MediaLive ignores the follow inputs. It finds the most recent fixed or immediate input that is in the past, relative to the UTC time at which you restart the channel. It starts ingesting the input at the start of the file.

  For example, assume the schedule looks like this:

  - Live input X with fixed start time of 11:00
  - File input A with fixed start time of 11:06
  - File input B with follow start time
  - File input C with follow start time
  - Live input D with fixed start time of 12:15

  Scenario 1: Assume the channel stopped at 11:04, when input X was active. You restart the channel at 12:09. The most recent fixed input switch relative to the current time is at 11:06. It is a switch to file input A. MediaLive goes to input A and starts ingesting that input from the beginning.

  Scenario 2: Assume the channel stopped at 11:04, when input X was active. You restart the channel at 12:16. The most recent fixed input switch relative to the current time is at 12:15. It is a switch to live input D. MediaLive goes to input D and starts ingesting.

  Scenario 3: Assume the channel stopped at 11:08, when input A was active. You restart the channel at 12:14. The most recent fixed input switch relative to the current time is at 11:06. It is a switch to file input A. MediaLive goes back to input A and starts ingesting. It ingests files A to C until 12:15, when it switches to the live input. It ingests at least part of file A. It might ingest files B and C. But at 12:15 it definitely switches to input D.

What happens with an empty schedule

If the channel finishes the last input in the schedule (so that the schedule is now empty) and you have set up so that the input doesn't loop, then MediaLive stops ingesting, but the channel continues to run. Charges for the channel continue to accrue.

Working with KLV metadata

You can configure MediaLive to pass through KLV metadata in TS outputs. The metadata must be compliant with SMPTE 336M-2007.

In an input, KLV metadata might be contained in a SMPTE 2038 stream or in a PID in a transport stream:

- If the KLV metadata is in a SMPTE 2038 stream in a specific input, you must configure the input to extract it. See the section called “SMPTE 2038 metadata”.
- If the KLV metadata is in a PID, read the topics that follow.

Note that if an AWS Elemental Link device is the input, the KLV metadata is always in a SMPTE 2038 stream. Therefore, read the SMPTE 2038 section.

Topics

Configuring inputs

When MediaLive ingests an input that contains a TS source, it automatically extracts KLV metadata that it finds. You don't need to configure the input.

Configuring outputs

You can choose to pass through the KLV metadata in one or more of the following output groups.

> **Note**
>
> The information in this section assumes that you are familiar with the general steps for creating a channel, as described in *Setup: Creating a channel* .

Archive

1. On the **Create channel** page, in the **Output groups** section, in the **Archive** group, choose the output.
2. In **Output settings**, select **Container settings**, then select **PID settings**.
3. Set these fields:
   - **KLV**: Choose **PASSTHROUGH**
   - **KLV data PIDs**: Enter the PID where you want the KLV metadata.

MediaPackage

MediaPackage outputs are automatically set up for passthrough. If MediaLive finds KLV metadata in an input, it passes it through in a MediaPackage output, in PID 501.

HLS

You can pass through KLV metadata in any output that has a standard HLS container (a TS container).

1. On the **Create channel** page, in the **Output groups** section, in the **HLS** group, choose the output.
2. In **Output settings**, make sure that **HLS settings** specifies **Standard HLS**.
3. In **HLS settings**, select **PID settings**.
4. Set these fields:
   - **KLV**: Choose **PASSTHROUGH**
   - **KLV data PIDs**: Enter the PID where you want the KLV metadata.

UDP/TS

1. On the **Create channel** page, in the **Output groups** section, in the **UDP** group, choose the output.
2. In **Output settings**, select **Network settings**, then select **PID Settings**.
3. Set these fields:
   - **KLV**: Choose **PASSTHROUGH**
   - **KLV data PID**: Enter the PID where you want the KLV metadata.

Working with AWS Elemental Link devices

For an overview of AWS Elemental Link devices, see the section called "AWS Elemental Link".

For information about working with a device, see *Setup: AWS Elemental Link* .

For information about monitoring a device, see *Operations: Monitoring devices*.

Implementing low latency outputs

You can create a glass-to-glass low latency workflow that uses AWS Elemental MediaLive and AWS Elemental MediaPackage. The channel in AWS Elemental MediaPackage must use MediaPackage v2.

> **Note**
>
> This section assumes that you are familiar with creating or editing a channel, as described in *Setup: Creating a channel* .

Follow these steps:

- Coordinate with the operator of the MediaPackage operator to obtain the destination URL. See the section called "HLS to MediaPackage v2".
- In the channel, create an HLS output group with MediaPackage v2 as the destination. Follow the guideance in the section called "Destination fields – MediaPackage".
- When you set up the outputs and the video stream in the output group, follow the guidance for these fields, to achieve optimum latency:

Section	Field	Description
HLS Settings, then CDN Settings	Connection Retry Interval	We recommend the same value as the segment length (in the **Manifest Segments** section). This value can affect latency.
	Num Retries	This value can affect latency.
	Filecache Duration	This value can affect latency. We recommend a lower number.
	Restart Delay	This value can affect latency.
Manifest Segments	Segment Length	We recommend 1 second for better latency.
	Min Segment Length	A value is required for delivery to MediaPackage. This value can affect latency.
HLS Output, then Settings then Gop Structure	GOP Size	This value can affect latency because the segment length is a function of the GOP size.
	Additional Settings > Closed GOP Cadence	This value can affect latency.

Customizing the paths inside HLS manifests

This section applies only to HLS outputs. Inside the HLS main manifest, there are paths to each child manifest. Inside each child manifest, there are paths to the media files for that manifest.

You can optionally change the syntax of these paths. Typically, you only need to change the syntax if the downstream system has special path requirements. Akamai CDNs usually require you to change the syntax.

Don't set up custom paths if the downstream system is MediaPackage. MediaPackage works with the default paths.

> **Note**
>
> The information in this section on HLS manifests assumes that you are familiar with the general steps for creating a channel, as described in *Setup: Creating a channel* .
> The key fields in the console that relate to this feature are in the **Location** grouping of the **HLS output group** section on the **Create channel** page. To review the step where you complete these fields, see the section called "The procedure".

Topics

Procedure to set up custom paths

The following fields relate to the paths inside the manifests:

- **HLS output group – Location** – the **Base URL manifest** fields
- **HLS output group – Location** – the **Base URL content** fields

To configure custom paths in manifests

1. Speak to the downstream system to find out if custom paths are required. The main manifests might need custom paths to the child manifests, the child manifests might need custom paths to the media files, or both main and child manifests might need custom paths. See the section called "How manifests work".
2. Design the paths, paying attention to the syntax and the rules for constructing the paths.

   See this guidance for different downstream systems.

See these examples.

3. Complete one or both of these fields in the **Location** section of the HLS output group page:

   - **Base URL manifest A** and **Base URL manifest B**. For a single-pipeline channel, complete only field A. For a standard channel, complete field A and field B.
   - **Base URL content A** and **Base URL content B**. For a single-pipeline channel, complete only field A. For a standard channel, complete field A and field B.

How manifests work

The following sections describe how manifest paths work.

How manifest paths work by default

The manifests that MediaLive creates include information about the paths to other files, specifically:

- The content inside the main manifest includes a path to each child manifest.

  By default, the syntax of this path is the following:

  ```
  baseFilename nameModifier extension
  ```

  For example:

  ```
  curling-high.m3u8
  ```

  The path is relative to the location of the main manifest.
- The content inside each child manifest includes a path to its media files.

  By default, the syntax of this path is the following:

  ```
  baseFilename nameModifier optionalSegmentModifier counter extension
  ```

  For example:

  ```
  curling-high-000001.ts
  ```

The path is relative to the location of the child manifest.

How custom paths work

If the default paths inside the manifests are not suitable for the way that the downstream system handles the three sets of files, you can complete the *base URL* fields:

- Complete the **Base URL manifest** fields so that MediaLive constructs custom paths to the child manifests.
- Complete the **Base URL content** fields so that MediaLive constructs custom paths to the media files.

When you customize the paths, the syntax changes.

- When you complete the **Base URL manifest** fields, the syntax for the child manifest path (inside the main manifest) is the following:

  ```
  baseURLManifest baseFilename nameModifier extension
  ```

  For example:

  ```
  http://viewing/sports/curling-high.m3u8
  ```

- When you complete the **Base URL content** fields, the syntax for the media file paths (inside the child manifests) is the following:

  ```
  baseURLContent baseFilename nameModifier optionalSegmentModifier counter
          extension
  ```

  For example:

  ```
  http://viewing/media/sports/curling-high-000001.ts
  ```

How MediaLive constructs these paths

The custom paths to the child manifests are constructed as follows:

- You complete the **Base URL manifest** fields, or the **Base URL content** fields, or both.

  For example:

  ```
  http://198.51.100/sports/viewing/
  ```

  Note the slash at the end of the value.
- MediaLive prepends that value to the default path. For example:

  ```
  http://198.51.100/sports/viewing/curling-high.m3u8
  ```

Rules for custom paths

Share the following rules with your contact person at the downstream system.

The general rule is that it's the responsibility of the downstream system to ensure that the custom paths work in their environment. MediaLive doesn't validate the values in any way. Therefore:

- If the protocol is specified (it is optional), it must be identical to the protocol that you specified in the **Destination URL** fields.
- The **Base URL manifest** and **Base URL content** fields for the same pipeline can have the same value or different values. They can be the same or different in any portion (the domain, path).
- The values can result in a relative path or an absolute path.
- A relative path to the child manifest is always relative to the location of the main manifest.
- A relative path to the media files is always relative to the location of the child manifest.
- The paths must end with a slash.

Guidance for setting up for custom paths

Following is some guidance for using the *base URL* fields for different downstream systems.

Setting up for custom paths if you control the downstream system

You might control the downstream system. For example, the downstream systems might be Amazon S3 or MediaStore connected to Amazon CloudFront. Your handling of the HLS files might require that you move one or more of the sets of files around. In this case, you could complete these *base URL* fields to match the paths of the final location of the files.

Setting up for custom paths if the downstream packager is MediaPackage

If the downstream package is MediaPackage, leave the **Base URL** fields empty. MediaPackage doesn't use this information.

Setting up for custom paths if you use a third-party downstream system

If you use a third-party downstream system, the downstream system must tell you whether to complete these **Base URL** fields.

Examples of custom paths

In all these examples, assume the following:

- In the main manifest, the default path to the child manifests is this relative path:

```
curling-high.m3u8
```

- In the child manifest, the default path to the media files is this relative path:

```
curling-high-000001.ts
```

Example 1

The downstream system is going to move the files from the location where MediaLive pushes them. The downstream system will move the files in such a way that the child manifests are still in the same relative location to the parent manifests, and the media files are still in the same relative location to the child manifests.

Therefore, you don't need to customize the paths. The default paths will still work after the move.

Example 2

You want the main manifest and the child manifests to include absolute paths to their respective files. You set up as follows:

- Complete the **Base URL manifest A** field to specify this absolute path:

```
http://198.51.100/sports/viewing/
```

  Inside the main manifest, the path to the child manifest will now be the following:

```
http://198.51.100/sports/viewing/curling-high.m3u8
```

- Complete the **Base URL content** field to specify this absolute path:

  ```
  http://203.0.113.55/sports/viewing/
  ```

  Inside the child manifests, the paths to the media files will now be the following:

  ```
  http://203.0.113.55/sports/viewing/curling-high-000001.ts
  ```

This example illustrates that the domain for the two sets of files could be different.

Example 3

You want the parent manifest to include absolute paths to the child manifests. But you want the child manifests to include paths to the media files that are relative to the child manifest. In this case, you customize the path to the child manifests, but you continue to use the default paths to the media files.

- You complete the **Base URL manifest A** field to specify this absolute path:

  Inside the main manifest for pipeline A, the path to the child manifest will now be the following:

  ```
  http://198.51.100/sports/viewing/curling-high.m3u8
  ```

- You don't complete the **Base URL content A** field.

  Inside the child manifests, the paths to the media files will still be the default:

  ```
  curling-high-000001.ts
  ```

Redundant HLS manifests

When you create an HLS output group in a standard channel, you can enable redundant manifests. Redundant manifests allow the downstream system (that reads the manifests) to better handle an output failure from MediaLive.

When the redundant manifest feature is enabled, the main manifest for each pipeline references both its own child manifests and the child manifests for the other pipeline. The downstream system finds the path to the child manifests for one pipeline. If there is a problem with that pipeline, then there will be a problem with the child manifests for that pipeline. The downstream system can then refer back to the main manifest to find the child manifest for the other pipeline. In this way, the downstream system can always continue with its processing of the manifest and media.

To successfully implement redundant manifests, you must be sure that the downstream system can handle redundant manifests in the ways that are described in the HLS specification.

> **Note**
>
> The information in this section on HLS manifests assumes that you are familiar with the general steps for creating a channel, as described in *Setup: Creating a channel* .
> The key fields in the console that relate to this feature are in the **Manifests and segments** grouping of the **HLS output group** section on the **Create channel** page. To review the step where you complete these fields, see the section called "The procedure".

Topics

Procedure to set up redundant manifests

To set up redundant manifests, you turn on the feature in the output group. You also make adjustments in the design of the output names and destination paths (compared to HLS outputs that don't implement redundant manifests).

The following field relates specifically to redundant manifests:

- **HLS output group – Manifests and Segments – Redundant manifests** field

To set up redundant manifests

1. Speak to the downstream system to find out if they support redundant manifests.
2. Read the information in the section called "Destination fields – HTTP server". Manifests are considered to be output from MediaLive. Therefore, the general rules about output destinations apply to redundant manifests.
3. Design the URLs for the two pipelines. There are special requirements for the URLs for the HLS files. Read the appropriate section:

   - the section called "Rules for most systems"
   - the section called "Rules for Akamai"

   These rules supplement the information in the section called "Destination fields – HTTP server".
4. If you also need custom paths for manifests, make sure you read the information in the section called "How custom paths work". You must consider the rules for custom paths when you design the URLs.
5. In the **HLS output group** section, for **Manifest and segments**, for **Redundant manifest**, choose **ENABLED**. This field applies to all outputs in the output group.
6. Complete these fields, following your design:

   - **Output group – HLS group destination** section
   - **Output group – HLS settings – CDN** section
   - **Output group – Location – Directory structure**
   - **Output group – Location – Segments per subdirectory**
   - **HLS outputs – Output settings – Name modifier**
   - **HLS outputs – Output settings – Segment modifier**

   - **HLS output group – Location –Base URL Manifest** (if you are also setting up custom paths)
   - **HLS output group – Location – Base URL Content** (if you are also setting up custom paths)

For information about how this feature changes the contents of the HLS manifests, see the section called "The media contents of an HLS manifest".

The results of this setup

Following is information about how redundant manifests work in three failure scenarios.

Scenario A – Input loss action is to emit output

If the input is lost on one of the pipelines and the **Input loss action** field is set to **EMIT_OUTPUT**, MediaLive continues to update the parent and child manifests.

From the point of view of the downstream system, there is no change to the parent or child manifests for either pipeline. The content inside the media files is filler content, but that doesn't affect how the downstream system reads the manifests.

Scenario B – Input loss action is to pause output

If the input is lost on one of the pipelines (for example, on pipeline 0) and the **Input loss action** field is set to **PAUSE_OUTPUT**, MediaLive does the following:

- It removes the listing for the child manifests for pipeline 0.
- It sends a request to the child manifest location for pipeline 0 to delete the child manifests.

The result for the downstream system that is reading the main manifest on pipeline 0: The system will no longer find a listing for the child manifests for pipeline 0. The system will look in the pipeline 0 main manifest for an alternative child manifest. If it finds the child manifest for pipeline 1, it will switch to reading that child manifest.

Downstream systems that are reading the main manifest for pipeline 1 are not affected because these systems are probably reading the child manifests for pipeline 1 (because these appear first in the manifest).

Scenario C – Pipeline failure

It is also possible for a pipeline to fail. This failure isn't the same as an input failure. When a pipeline fails (for example, pipeline 0), the following happens:

- Output stops.
- The main manifest for pipeline 0 doesn't get deleted. It still contains a listing for the child manifests for pipeline 0.
- The child manifests are not updated because no new media files are being produced. The child manifests are *stale*.

- The main manifest for pipeline 1 doesn't change. It still contains a listing for the child manifests for pipeline 0 (and for pipeline 1).

The result for the downstream system that is reading the main manifest for pipeline 0: The system will find a listing for child manifests for pipeline 0, but that manifest will be stale. If the system can detect that the manifest is stale, it can return to the pipeline 0 main manifest and search for an alternative child manifest. If it finds the child manifest for pipeline 1, it will switch to reading that child manifest.

Downstream systems that are reading the main manifest for pipeline 1 are not affected. These systems are presumably reading the child manifests for pipeline 1 (because these appear first in the manifest).

> **Note**
>
> If the downstream system for the HLS output is AWS Elemental MediaStore, you can set up MediaStore to delete stale inputs. See Components of an object lifecycle policy. After the child manifest has been deleted, MediaStore falls back to following the "manifest has been deleted" logic of scenario B.

The media contents of an HLS manifest

Setting up redundant manifests changes the contents of the HLS manifest. It changes the media information (the video, audio, and captions information) inside the manifests. All of this information appears as `#EXT-X-STREAM-INF` tags.

The following sections describe the number of these tags and the contents of these tags in a standard (not redundant) manifest and in a redundant manifest.

What a standard manifest looks like

With a standard channel, there are two pipelines. Each pipeline produces its own set of manifests. Therefore, for pipeline 0, there is one main manifest, one set of child manifests, and one set of media files. Similarly, pipeline 1 has the same set of files. The manifests reference only the files for their own pipeline.

The video information in the main manifest for each pipeline might look like this:

```
#EXT-X-STREAM-INF:BANDWIDTH=629107 ...
curling-high.m3u8
```

What a redundant manifest looks like

When the redundant manifest feature is enabled, each main manifest references the child manifests for its own pipeline and for the other pipeline.

This feature doesn't affect child manifests. Child manifests only reference their own media files.

Following is an example of how the video information in the manifest might appear. Assume that the baseFilename for pipeline 0 is *first-curling* and for pipeline 1 it is *other-curling*.

The manifest for pipeline 0 might look like this (with the child manifest information for pipeline 0 appearing first):

```
#EXT-X-STREAM-INF:BANDWIDTH=629107 ...
first-curling-high.m3u8

#EXT-X-STREAM-INF:BANDWIDTH=629107 ...
other-curling-high.m3u8
```

The video information in the manifest for pipeline 1 might look like this (with the child manifest information for pipeline 1 appearing first):

```
#EXT-X-STREAM-INF:BANDWIDTH=629107 ...
other-curling-high.m3u8

#EXT-X-STREAM-INF:BANDWIDTH=629107 ...
first-curling-high.m3u8
```

Rules for most downstream systems

Read this section if you are setting up redundant manifests with any downstream system except Akamai. If your downstream system is an Akamai CDN, see the section called "Rules for Akamai".

The rules for downstream systems (except Akamai) are based on these requirements:

- MediaLive pushes the files from both pipelines to the same location (protocol/domain/path).

- Given that the location is the same, the base filenames for the pipelines must be different.
- If you are also implementing custom manifest paths, the URL inside the manifests must be identical.

Make sure that the downstream system can work with these restrictions.

Field	Rule
Protocol/domain/path portion of the two destination URIs (A and B)	Must be identical in both fields.
Base-filename portion of the two destination URIs (A and B)	Must be different in each field. It *cannot* use variable identifiers that include the date or time.
NameModifier for each output	There is only one instance of this field. Both pipelines use the same value. It *cannot* use variable identifiers that include the date or time.
Segment modifier	There is only one instance of this field. Both pipelines use the same value. It *can* use variable identifiers that include the date or time.
Base URL Manifest A and Base URL Manifest B	These fields apply only if you are also implementing custom manifest paths. Complete both fields.
Base URL Content A and Base URL Content B	These fields apply only if you are also implementing custom manifest paths. Complete both fields.

Rules for Akamai CDNs

Read this table if you are setting up redundant manifests with an Akamai CDN. If your downstream system is not an Akamai CDN, see the section called “Rules for most systems”.

Field	Rule
Protocol/domain/path portion of the two destination URIs (A and B)	Can be different from each other, or can be the same.
BaseFilename portion of the two destination URIs (A and B)	Can be different from each other, or can be the same. It *cannot* use variable identifiers that include the date or time. The combination of the protocol/domain/path and the baseFilename must be unique in A and B. This rule ensures that the output files from the two pipelines don’t overwrite each other.
Name modifier	There is only one instance of this field. Both pipelines use the same value. It *cannot* use variable identifiers that include the date or time.
Segment modifier	There is only one instance of this field. Both pipelines use the same value. It *can* use variable identifiers that include the date or time.
Base URL Manifest A and Base URL Manifest B	These fields apply only if you are also implementing custom manifest paths. Typically, with Akamai CDNs, you do implement custom manifest paths.

Field	Rule
	Complete both fields.
Base URL Content A and Base URL Content B	These fields apply only if you are also implementing custom manifest paths. Complete both fields.

Combining redundant manifests with other features

Combining redundant manifests and custom path feature

You can set up custom paths in redundant manifests. Make sure you follow the rules for custom paths and for redundant manifests for your downstream system—either an Akamai CDN or another downstream system.

Combining redundant manifests with audio rendition groups

> **Note**
>
> The information in this section assumes that you are familiar with the manifests for audio rendition groups. For more information, see the section called "Sample manifest".

If you have set up redundant manifests and you have an audio rendition group, MediaLive automatically adjusts the references to the audio rendition groups in the parent manifests.

In each pair of lines (for example, the `#EXT-X-STREAM-INF` for the high-resolution video), MediaLive adjusts the name of the rendition groups. In this way, the references to the rendition groups are different for each pipeline, which ensures that when the client player reads the manifest, it chooses the video and the audio from the same pipeline.

The `#EXT-X-STREAM` for the video for pipeline 0. Note the value for *AUDIO*:

```
#EXT-X-STREAM-INF:BANDWIDTH=541107,...AUDIO="aac-audio-0", ...
```

The `#EXT-X-STREAM` for the video for pipeline 1. Note the value for *AUDIO*:

```
#EXT-X-STREAM-INF:BANDWIDTH =541107, ...AUDIO="aac-audio-1",...
```

Working with metadata in AWS Elemental MediaLive

MediaLive supports several types of metadata:

- ID3 metadata
- KLV metadata
- SMPTE 2038 metadata

Working with motion graphics overlays

You can use the motion graphics overlay feature to superimpose a motion image onto the video in a MediaLive channel. The motion image is based on an HTML5 motion graphic asset.

To set up for motion graphics overlay, you must perform work in two areas:

- You must choose an *HTML5 authoring system*. You must use this authoring system to prepare an HTML5 asset, and you must continually publish the asset to a location outside of MediaLive.
- On MediaLive, you must enable motion graphics in each channel where you want to include a motion graphic overlay.

After you have started the channel, you use the schedule feature in MediaLive to insert the motion graphic in the running channel. As soon as the schedule receives the action, MediaLive starts to download and render the content. It continually downloads and renders the content for as long as the motion graphics action is active. At any time, you can deactivate the image by creating a deactivate action in the schedule.

Pricing

There is a charge for running a channel that has the motion graphics overlay feature enabled. There is a charge even when there is no motion graphics overlay currently inserted in the channel.

The charge is based on the largest video output in the channel.

To stop this charge, you must disable the feature.

For information on charges for using this mode, see the MediaLive price list. https://aws.amazon.com/medialive/pricing/

Topics

Step 1: Prepare the motion graphic asset

You use an authoring system to create the asset and to manage the content, including implementation of features such as fade or opacity.

MediaLive's role in displaying the graphics overlay is limited to rendering the asset, and to inserting it and removing it from the video at the specified times. MediaLive doesn't provide any features for manipulating the motion graphic.

To prepare the motion graphic asset

1. Use the authoring system to create the asset. The HTML5 content must meet these requirements:
   - It can be any HTML5 authoring system that uses standard browser-based rendering techniques.
   - It can use any HTML5 tags except video and audio.
   - It can incorporate Javascript to interact with a backend system that provides the ability to dynamically control the asset that is being published to the source URL.
   - You should size the content to match the width and height of the largest video rendition in your channel. MediaLive can't change the resolution of the asset to fill the frame, although it will resize content down to fit a smaller video rendition without cropping.
2. Publish the motion graphic asset to a source URL that is accessible via a public IP address.
3. Make a note of the location. You will need it when you add the schedule action.
4. If the location of the motion graphics asset requires login in order to download files, obtain the required user name and password. Make a note of the credentials. You will need them when you add the schedule action.

Step 2: Enable the feature

Perform this step for each channel where you want to insert a motion graphic overlay.

> **Note**
>
> The information in this section assumes that you are familiar with the general steps for creating a channel, as described in *Setup: Creating a channel* .

To enable the feature

Follow this procedure when you create the channel or when you are modifying an existing channel.

1. Display the **General channel settings** section and choose the **Motion graphics configuration** pane.
2. Turn on the **Enable motion graphics configuration**. More fields appear.
3. Set the fields as follows:

   - **Motion graphics insertion** – Set to **Enabled**.
   - **Motion graphics settings** – Leave the value as **HTML motion graphics** (the only option).

   For detailed information about this section of the **Create channel** page on the console, see the section called “General settings”.

   When you create or save the channel, it will be set up for motion graphics overlay. This means that when you add an action to the channel schedule, the option for motion graphics will appear in the list in the **Action type** field.

> **Important**
>
> When you save a channel that has the motion graphics overlay feature enabled, there is a charge for the feature that applies when the channel is running, even when there is no motion graphics overlay currently being inserted.
> To stop this charge, you must disable the feature.

To disable the feature

To disable the motion graphics feature, turn off the **Enable motion graphics configuration** field.

Step 3: Insert the overlay

When you are ready, you can create an action in the channel schedule to activate (insert) the overlay. You can create the action at any time – before the channel has started or while it is already running.

The schedule is a timetable that is attached to each channel. The schedule is designed to let you specify actions to perform on the channel at a specific time. You can set up the action so that a motion graphic is active for a specific time, or so that it is active indefinitely. In both cases, you can stop the overlay at any time by creating a deactivate action.

For detailed information, see *Setup: Creating a schedule* and the section called "Creating actions".

Multiplex and MPTS in AWS Elemental MediaLive

You can set up a MediaLive multiplex to create a multi-program transport stream (MPTS). You might be interested in MediaLive multiplex if you are a service provider who has experience in distributing transport stream (TS) content over RTP or UDP.

> **Note**
>
> The term *MediaLive multiplex* refers to an entity in MediaLive. The term MPTS is a standard term in digital transmission technology. You create and work with a MediaLive multiplex in order to create an MPTS for distribution.

Topics

Overview of multiplex and MPTS in AWS Elemental MediaLive

A multi-program transport stream (MPTS) is a UDP transport stream (TS) that carries multiple programs. AWS Elemental MediaLive lets you create an MPTS that contains all variable bitrate programs, a mix of variable and constant bitrate programs, or all constant bitrate programs.

To create an MPTS, you create a MediaLive multiplex. You then add up to 20 MediaLive programs to the multiplex. Finally, you create one MediaLive channel for each program, and associate each channel with its program.

Channel

The channel is a regular MediaLive channel that is configured in a specific way. The channel is dedicated to a multiplex, which means that you can't use it to produce both an MPTS output and other outputs (such as SPTS UDP or HLS outputs).

Supported sources are those that use a MediaConnect input or an MP4 input.

The channel contains only one output group, of type **Multiplex**, and one output. This output is a transport stream. Apart from these special requirements for the input and output, the channel is like any regular channel. For the video, audio, and captions that it produces, it follows the rules for a UDP output.

The channel is always a standard channel. It can include any of the regular channel features that you can implement for a UDP output, such as input switching and SCTE-35 ad avails messages.

Program

The channel is attached to a MediaLive program.

The program provides information about the bitrate for the video in this program. Each program can have a constant video bitrate, or it can have a variable video bitrate. For a variable video bitrate, the multiplex allocates the bitrate for the program based on the demands of all the programs.

Multiplex

Each program is attached to the multiplex. A multiplex can contain up to 20 programs.

The MediaLive multiplex provides configuration information for the MPTS, including the bitrate of the entire MPTS.

Starting a Multiplex

When you are ready, you start the multiplex and the channels. (You don't start the programs.)

The MPTS is an RTP output. MediaLive creates and delivers the MPTS to AWS Elemental MediaConnect in the account associated with the MediaLive that is creating the MPTS. AWS Elemental MediaConnect automatically sets up the RTP output as an entitled source. You don't have to perform any steps to set up this entitled source. But in order to complete the distribution of the MPTS, you must create a flow that uses that entitled source.

For more information about starting the multiplex, see the section called "Starting, pausing, or stopping a multiplex". For more information about entitled sources, see Creating a Flow in the *AWS Elemental MediaConnect User Guide*.

Restrictions for multiplexes

Following is a summary of the restrictions associated with multiplexes:

- There are service quotas for the number of multiplexes you can create. For more information, see *Quotas*.
- These limitations apply to a multiplex:
  - Each multiplex produces only one MPTS. The MPTS has two pipelines, so it is sent to two destinations.
  - All multiplex outputs must include video.
- These limitations apply to a program:
  - Each program in a multiplex is single use. It is attached only to one multiplex, and you can use it only for that multiplex.
- These limitations apply to a channel in a multiplex:
  - Each channel is single use. You can attach it to only one program in the multiplex, and you can use it only for that multiplex.
  - Each channel contains one and only one output group, of type multiplex. It can't contain any other type of output group.

Setting up a multiplex

There are three components involved in an MPTS: a MediaLive multiplex, MediaLive programs, and MediaLive channels (and their attached MediaLive inputs). You must create these components in this order:

- Create the MediaLive multiplex.
- Create programs in that multiplex. A program can't exist on its own; it always exists in a multiplex.
- Create one channel and attach it to the program. A multiplex channel can't exist on its own; it always exists in a program.

Step 1: Plan the availability zones

Identify two AWS Availability Zones for the multiplex. AWS Elemental MediaLive runs the pipelines for the multiplex in those two zones. Follow these guidelines:

- If the multiplex will include a MediaConnect input and that input already exists, then make a note of the region and Availability Zones of the flows in that input. In the steps below, you will set up the multiplex to use the same region and Availability Zones.
- If the multiplex will include a MediaConnect input and that input doesn't already exists, then decide choose a region and Availability Zones. The flows and the multiplex must use the same region and Availability Zones.
- If the multiplex won't include a MediaConnect input, then choose a region and Availability Zones for the two pipelines in the multiplex.

Step 2: Create the multiplex

Create the multiplex. Make sure to create the multiplex in the identified region and Availability Zones. For more information, see the section called “Creating a multiplex and program”.

Step 3: Create the inputs

You must create the inputs for the channels that you will create. As with any channel, you must create the inputs before you create each channel.

- Follow the regular procedure for creating the input.
- Inputs for the channels that are used in a multiplex can be MP4 inputs or MediaConnect inputs.
- For MediaConnect inputs, make sure that you follow these rules:
  - The flows in the MediaConnect inputs must use the region and zones that you identified in step 1.
  - All the MediaConnect inputs must use these same two zones.

Step 4: Create the programs

Create the programs to add to the multiplex. For more information, see the section called "Creating a multiplex and program". You can add up to 20 programs per multiplex. The multiplex must already exist.

Step 5: Create the channels

Create a channel for each program. The program must already exist.

Using the console, there are two ways to create the channel for a program:

- From the **Program** details page. After you create each program, details about the program appear, including a link to immediately create a channel for the program. If you choose this link, the **Create channel** page appears, with many fields already set to the value that is applicable to a channel used in a multiplex. For a summary of the fields that MediaLive sets for you, see the section called "Restrictions".
- From the navigation pane. You can create a channel in the usual way, by choosing **Channel** from the navigation pane. For information about setting some of the fields, see the section called "Restrictions".

For more information about completing the channel fields, see *Setup: Creating a channel* .

Restrictions

There are some restrictions on the configuration of a channel that is used in a multiplex:

Restrictions in the Output Group

The channel can contain only one output group, of type **Multiplex**. This type follows the rules of a UDP output group. It can contain only one output.

Restrictions in the Output

The following restrictions apply to the output fields.

Field	Value
In **Multiplex destination**, the **Multiplex program** field	From the list, choose the multiplex program that this channel belongs to.

Field	Value
In **Stream settings**, for **Video**	The output can contain one, and only one, video asset.
In **Stream settings**, for **Audio**	The output can contain zero or more audio assets.
In **Stream settings**, for **Captions**	The output can contain zero or more captions assets.

Restrictions in the Video

The following rules apply to the fields in the video.

Field	Value
Width and **Height** (resolution)	Set values for both the width and height. The width can be up to1920 pixels. The height can be up to1080.
Codec settings	Choose **H.264** (AVC) or **H.265** (HEVC).
In **Aspect Ratio**, the **PAR control** field	Set a value. This is required. Don't set up to follow the aspect ratio from the source.
In **Rate control**, the **Rate control mode** field	Choose **Multiplex**.
In **Rate control**, the **Buffer size** field	Keep blank.
In **Frame rate**, the **Framerate** field	Set a value. This is required. Don't set up to follow the frame rate from the source. The numerator and denominator must result in a decimal value in this range: • Lowest supported rate is 23.97 frames per second (2400/1001).

Field	Value
	• Highest supported rate is 60 frames per second.
In **GOP structure**	For **GOP size units**, choose **FRAMES**. Then set **GOP structure** to 6 or greater. Or for **GOP size units**, choose **SECONDS**. Then set **GOP structure** to 0.1 or greater.
In **Codec details**, the **Profile** field	If the codec is H.264, choose one of these profiles: • **BASELINE** • **HIGH** • **MAIN** If the codec is H.265, choose one of these profiles: • **BASELINE** • **HIGH** • **HIGH_10BIT** • **MAIN**

Features That Are Not Restricted

There are some features of the channel that you can set up in the same way as you would set them up in a regular channel:

- For video configuration fields not mentioned in the table earlier in this section, you can set the field to suit your workflow.
- For audio, you can set up as you would set up in a UDP output group in a regular channel.
- For captions, you can set up as you would set up in a UDP output group in a regular channel. Specifically, make sure that the input captions and output captions follow the rules for a UDP output group. See the section called “Captions: Supported formats”.

- For other features, if the feature is available for a UDP output group, then it is available for a channel in a multiplex.

Starting, pausing, or stopping a multiplex

At runtime, you start both the multiplex and the channels in the multiplex. You can stop the multiplex and the channels independently of each other. You don't start or stop a program. (You can't perform any actions on a program except for create and delete.)

Topics

Summary of these actions

The following table summarizes the start, stop and pause capabilities for the multiplex, program, and channel.

Item	Action	Note
Multiplex	Start	You can start a multiplex and the channels in any order.
	Stop	You can stop a multiplex and leave the channels running. But there is no operational reason to stop a multiplex . You can edit the multiplex without stopping it.
	Pause	You can't pause a multiplex.
Program	Any	You don't start or stop a program.

Item	Action	Note
Channel	Start	You can start a channel that is used in a multiplex at any time, including before you have started the multiplex.
	Stop	You can stop a channel without stopping the multiplex. You must stop a channel in order to edit it.
	Pause	You can't pause a channel that is used in a multiplex.

Starting the multiplex

To start streaming the MPTS, start the multiplex and the channels. You can start the channels and then start the multiplex. Or you can start the multiplex and then start the channels.

If any channels are multi-input channels, the standard recommendations about starting and restarting these channels apply. For more information, see the section called “Starting and restarting the channel”.

Contents of the MPTS

After you start the multiplex and channels, MediaLive starts all these components. MediaLive creates two multiplex pipelines, each of which creates a separate MPTS asset. The MPTS contains the following:

- The SDT contains an entry for each program.
- The PAT contains an entry for each program that has a MediaLive channel associated with it.
- The PMT for each program contains an entry for each stream that is being used. When you created the programs, MediaLive allocated the PIDs for all possible program streams. At runtime, the PMT references only those PIDs that actually contain content.
- One PID for each stream.

If you add or remove programs and channels while the multiplex is running, or if you modify channels while the multiplex is running, MediaLive modifies the MPTS tables dynamically.

Encoding

MediaLive encodes the content in each channel in the regular way, except that the MediaLive multiplex continually communicates with each MediaLive channel to provide a bitrate for each video segment. The MediaLive multiplex creates an MPTS from the output of all the channels.

Distribution

The MPTS is an RTP output. MediaLive creates and delivers the MPTS to AWS Elemental MediaConnect in the account associated with the MediaLive that is creating the MPTS. AWS Elemental MediaConnect automatically sets up the RTP output as an entitled source. You don't have to perform any steps to set up this entitled source. But in order to complete the distribution of the MPTS, you must create a flow that uses that entitled source.

The entitlement name includes the string "multiplex" and the multiplex ID, so that MediaConnect users can easily identify it.

For more information about entitled sources, see Creating a Flow in the *AWS Elemental MediaConnect User Guide*.

Pausing activity in the multiplex

You can't pause a multiplex. You can only stop it. You also can't pause a channel used in a multiplex. This rule applies even though you can pause a regular channel.

Stopping activity in the multiplex

You can stop a multiplex or a channel.

Stopping a multiplex

Typically, after your multiplex is in a production environment, you stop the multiplex only to delete it. You don't need to stop the multiplex to modify it, except to modify the **Maximum Video Buffer Delay** field.

When you stop a multiplex, the channels continue to run, although their outputs are not with an MPTS, so the outputs don't go to their destinations.

When you stop a multiplex, you stop accruing charges for the multiplex. But you still accrue charges for the channels in the multiplex, unless you also stop those channels.

To stop a multiplex

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Multiplexes**, and then choose the multiplex that you want to stop.
3. On the **Details** pane, choose **Multiplex actions**, and then choose **Stop multiplex**. If there are programs and those programs have running channels, then the channels continue to run, although their outputs don't go to their destinations.

Stopping a channel in a multiplex

You must stop a channel to change its configuration or to delete it.

When you stop a channel, the multiplex continues to run. MediaLive modifies the PMT to remove the PAT for the associated program.

When you stop a channel, you stop accruing charges for the channel. But you still accrue charges for the multiplex, unless you also stop the multiplex. You should review the charges for a running multiplex; you might consider that there is not a lot of gain in stopping the multiplex.

To stop a channel

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Multiplexes**, and then choose the program for the channel.
3. On the **Programs** pane, choose the program or programs, choose **Multiplex actions**, and then choose **Stop channel**.

You can also stop a channel in the multiplex in the same way as you stop a regular channel. For more information, see *Operations: Start, stop, and pause channel*.

Nielsen watermarks

You can set up AWS Elemental MediaLive to create new Nielsen watermarks and insert them in the output audio. Typically, only content and distribution providers use Nielsen watermarks. If you're

not working with The Nielsen Company to implement watermarks, you don't need to read this section.

If your content already contains watermarks, you might choose to convert them to ID3 metadata and include that metadata in the output. For more information about passthrough and conversion to ID3, see the section called “Nielsen watermarks to ID3”.

Topics

Audio requirements

Supported audio

Your audio must meet the following requirements:

- Sample-rate frequency: 48 kHz (48000 samples per second).
- Up to 8 audio channels, with interleaved samples.
- The audio must conform to one of the coding modes and channel layouts specified in the following table.

In the table, read across each row to identify the channel layout for the coding mode that is identified in the first cell.

Number of channels	Coding mode	Ch 1	Ch 2	Ch 3	Ch 4	Ch 5	Ch 6	Ch 7	Ch 8
1	Mono	Left							
2	Stereo	Stereo left	Stereo right						
6	5.1 audio	Front left	Front right	Center	LFE	Surrounc left	Surrounc right		

Number of channels	Coding mode	Ch 1	Ch 2	Ch 3	Ch 4	Ch 5	Ch 6	Ch 7	Ch 8
8	5.1 audio plus stereo	Front left	Front right	Center	LFE	Surround left	Surround right	Stereo left	Stereo right

Recommended minimum bitrates

We strongly recommend following the minimum audio bitrates listed in the following table. If you set the audio bitrates lower than the recommended values, your watermarks might not be reliably detected.

Codec	Coding mode	Minimum bitrate (kbps)
Dolby Digital	Stereo	192
	5.1	384
Dolby Digital Plus	Stereo	192
	5.1	192
AAC with the LC profile	Stereo	128
AAC with the HEV1 profile	5.1	256
MPEG-1, layer II	Stereo	96

Getting ready

To get ready for watermarks

1. Determine if you should insert NAES II (N2), NAES VI (NW) watermarks, or CBET watermarks. NAES II are used in the United States. CBET are used in Canada. You can insert one or both types in the same audio encode.

2. Obtain the following information from your contact at The Nielsen Company:

   - For NAES II or NAES VI watermarks:
     - Source Identification Code (SID).
     - NAES check digit code.
   - For CBET watermarks:
     - CBET Source Identification (CSID) code.
     - CBET check digit code.

   You must obtain separate sets of values for each channel.
3. If you are setting up CBET watermarks, decide how you want to handle watermarks that are already in the source audio. The options are the following:

   - Remove all the existing watermarks and replace them with new ones.
   - Keep the existing watermarks. MediaLive will insert new watermarks only in portions of the audio stream where there are no watermarks.

Setting up Nielsen watermarks

> **Note**
>
> The information in this section assumes that you are familiar with the general steps for creating a channel, as described in *Setup: Creating a channel* . It also assumes that you have already set up the audio encodes (outputs) that will contain the watermarks.

To create Nielsen watermarks

1. On the **Create channel** page of the MediaLive console, choose the output group in the left navigation bar. Then choose the output. In the **Stream settings** pane, choose the desired **Audio** tab.

   In the **Codec settings** section, expand the **Additional encoding settings** section. Expand the **Additional settings** drop-down menu, and find the **Audio Watermark Settings** field.
2. Choose **Audio watermark**, then choose **Nielsen Distribution Type**.
3. Choose the option that applies:

- **Program content**: Typically, this option applies if your organization is a network broadcaster.
- **Final distribution**: Typically, this option applies if your organization is a broadcast affiliate or a cable network provider.

4. If you want to include CBET watermarks: In the **CBET Settings** field, choose **Nielsen CBET**. More fields appear.

   If you want to include NAES watermarks: In the **NAES II and NW Settings** field, choose **NAES II and NW**. More fields appear.

   You can include both sets of watermarks in the same output audio.

5. Complete the fields as shown in the table.

Area	Field	Description
CBET Settings	**CBET Source ID (CSID)**	The value that you obtained from Nielsen.
	CBET Check Digits	The value that you obtained from Nielsen.
	CBET Stepaside	**ENABLED**: Nielsen watermarks that are already in the source are left intact. MediaLive will insert new watermarks only in portions of the audio where there are no existing watermarks. **DISABLED**: Existing Nielsen watermarks are removed. MediaLive will insert new watermarks throughout the audio.
NAES II and NW Settings	**Source ID (SID)**	The value that you obtained from Nielsen.

Area	Field	Description
	Check Digits	The value that you obtained from Nielsen.
	Timezone	Choose a time zone to apply to the watermarks. Or leave as UTC, which is the default.

Converting Nielsen watermarks to ID3

If one or more inputs in a channel includes Nielsen watermarks in the audio, you have the option of setting up the channel to convert those watermarks to ID3 metadata. These watermarks are part of the measurement and analytics capabilities supported by Nielsen.

This option applies only in the following scenario:

- One or more inputs in your channel includes Nielsen watermarks in the audio.
- Your channel has at least one output group that can include the Nielsen ID3 tag:
  - Archive output group
  - CMAF Ingest output group
  - HLS output group. The output must be a standard output (not an audio-only output)For example, an HLS output group.
- You know that at least some of your playback devices implement the Nielsen SDK. This SDK provides functionality to handle the ID3 tags.

Converting the watermarks to ID3 tags doesn't remove the original watermarks. Outputs where you include the ID3 tags will contain both the watermark and the ID3 tags. Outputs that don't include the ID3 tags will contain only the watermark.

You can't remove the watermarks from the audio, but if your playback devices don't implement the Nielsen SDK, the devices simply ignore the watermarks.

> **Note**
>
> Do not confuse this feature with the ability to insert ID3 metadata in outputs.

To set up watermarks as ID3 tags

1. On the **Create channel** page, in the **General settings** section, in the **Nielsen Configuration** pane, choose **Enable Nielsen configuration**.
2. Set the fields as follows:
   - **Nielsen PCM to ID3 tagging**: Choose **ENABLED**.
   - **Distributor ID**: Optionally, enter the distributor ID that you obtained from Nielsen. If you enter an ID here, it is added to the ID3 metadata along with the source ID (SID) that is always in the source watermark.
3. Go to the output group and output where you want to include the ID3 tags..

   (If the output group is **MediaPackage**, you don't have to set up the output. The ID3 tags are always passed through, if the output is a standard output.)

Output group	Section	Instruction
Archive	**Output settings**	Choose **PID settings**. In **Nielsen ID3**, choose **PASSTHROUGH**.
CMAF Ingest	**CMAF Ingest settings**	In **Nielsen ID3 Behavior**, choose **PASSTHROUGH**.
HLS	**Output settings**	The contain must be a standard HLS container. Verify the value in the **HLS Settings** field.

Output group	Section	Instruction		
		Choose **PID settings**. In **Nielsen ID3 behavior**, choose **PASSTHROUGH**.		
UDP	**Output settings**	Choose **Network settings**, then choose **PID settings**. In **Nielsen ID3**, choose **PASSTHROUGH**.		

Implementing pipeline locking

The pipeline locking feature ensures that the output from the two pipelines in a standard channel are frame-accurate with each other. Pipeline locking applies only to specific output types.

There are two modes of pipeline locking:

- Pipeline locking (the default): lock the two pipelines to each other
- Epoch locking: lock the pipelines using the Unix epoch as the reference.

You can't disable pipeline locking in the applicable output types. But you should configure the behavior, to make sure it suits your workflow.

> **Note**
>
> You might be familiar with the term *output locking*. In MediaLive, the term used is *pipeline locking*. Whatever term is used, the effect is identical: frame accurate outputs.

Topics

Requirements for pipeline locking

Topics

Channel requirements

The channel must be a standard channel. The channel can have multiple inputs attached. In other words, the channel can implement input switching.

Output types

Pipeline locking works only with the following types of output groups:

- HLS
- MediaPackage
- Microsoft Smooth
- UDP

The channel can contain output groups other than the supported ones. MediaLive will not perform pipeline locking on those output groups. This means that in those other output groups, there is no guarantee that the two pipelines will be frame-accurate with each other.

Epoch locking and SCTE 35

There are constraints for using epoch locking in an HLS or MediaPackage output group.

HLS output group

It's not possible to enable SCTE 35 passthrough or manifest decoration in an HLS output group in a channel that uses epoch locking. You will get a validation error when you save the channel. You must decide how to resolve this conflict:

- Don't enable epoch locking in the entire channel: You can set the mode to regular pipeline locking in the entire channel, and keep SCTE 35 passthrough in the HLS output group.
- Disable SCTE 35 passthrough in the HLS output group: You can keep epoch locking but disable SCTE 35 passthrough and manifest decoration in the HLS output group. You can still enable SCTE 35 passthrough in other output groups.

MediaPackage output group

For a MediaPackage output group, constraints apply if the input includes SCTE 35 messages:

- When epoch locking isn't enabled in the channel, MediaLive automatically passes through any SCTE 35 messages from the input and automatically enables manifest decoration.
- When epoch locking is enabled, MediaLive automatically disables SCTE 35 passthrough and manifest decoration in the MediaPackage output group.

You should decide which feature you want to keep. You can keep the SCTE 35 messages (in which case you must disable epoch locking in the entire channel). Or you can enable epoch locking but lose passthrough of the SCTE 35 messages. Note that there is no advantage to setting up the output as an HLS output group, because similar constraints apply, as described above.

Step 1: Verifying the input

You must speak to your upstream system to make sure that every input attached to the channel meets these requirements:

Supported input types

The channel can't include HLS inputs.

Requirement for embedded timecode

The input must include embedded timecode. These rules apply:

- For both locking modes, the input must have an embedded timecode.
- For epoch-locking mode, the embedded timecode must be within 2 minutes of epoch time. If the timecode is off by more than 2 minutes, MediaLive considers that the source doesn't meet the requirements for pipeline locking.

Frame rate requirements

The conversion between the input framerate (or framerates) and the desired output framerate must be *simple*, which means that one of these statements must apply:

- The output framerate must be a whole number multiple of the input framerate. For example, the input framerate might be 45 FPS, and the output framerate might be 90 FPS.
- The input framerate must be a whole number multiple of the output framerate. For example, the input framerate might be 60 FPS, and the output framerate might be 30 FPS.

Note that with these rules, it is possible for the framerates to be integers. For example, if the input framerate is 29.97 FPS and the output framerate is 59.94 FPS.

Following are examples of *complex* framerates. You *can't* use the input if one of these combinations applies to your channel:

- Input FPS is 59.4, output FPS is 60.
- Input FPS is 45, output FPS is 60.
- Input FPS is 29.97 FPS, output FPS is 23.978.

Step 2: Setting up for locking

Pipeline locking is always enabled. But you must set up the channel to make sure that MediaLive can successfully perform pipeline locking in your output groups.

> **Note**
>
> All the procedures in this section assume that you are familiar with the general steps for creating a channel, as described *Setup: Creating a channel* .

Setting the mode

You can configure the channel to perform pipeline locking using one of these modes:

- Pipeline locking: lock the two pipelines to each other
- Epoch locking: lock the pipelines using the Unix epoch as the reference.

Configure the pipeline locking mode

1. In the channel that you are creating, in the navigation pane, choose **General settings**. Then choose **Global configuration**.
2. Choose **Enable global configuration**.
3. In **Output locking mode**, choose the mode—**PIPELINE_LOCKING** or **EPOCH_LOCKING**. For details about the options, choose the **Info** link next to the field.

Setting up an HLS, MediaPackage, or Microsoft Smooth output group

In an HLS output group or Microsoft Smooth output group, you must set up the framerate for each video encode.

Set up for pipeline locking

1. In the channel that you are creating, in the navigation pane, choose the HLS or Microsoft Smooth output group. If necessary, create the outputs and video encodes in each output.
2. In each output that contains a video encode, choose the video encode. In the **Codec settings** field, choose the codec. More fields appear.
3. Choose the **Frame rate** section and set the following fields:

   - **Framerate control**: We recommend you choose **Specified**. The option **Initialize_from_source** doesn't work well with pipeline locking.

- **Framerate numerator** and **Framerate denominator**: Set the desired resolution for the output. Make sure that the conversion from input framerate to output framerate meets the requirements.

4. Repeat, to setup of the frame rate in the video encode in every output.

Setting up a UDP output group

In a UDP output group, you must obtain information about segmentation markers, and set up the segmentation markers for framerate for each video encode.

Set up for pipeline locking

1. You need information about the how to configure segmentation in the outputs. This information is contained in fields on the **Create channel** page on the console. To display the fields, in the navigation pane choose **Archive group**. Then choose an output and choose **Network settings**. Choose the **Info** link next to each of the following fields:

   - **Segmentation markers**
   - **Segmentation time**
   - **EBP lookahead msec**
   - **Fragment time**
   - **Segmentation style**
   - **EBP placement**
   - **EBP audio interval**

2. Speak to your contact at the downstream system to obtain recommended values for these fields.
3. In the channel that you are creating, in the navigation pane, choose the Archive output group. If necessary, create the outputs. Then in the **Output settings**, choose **Network settings**. More fields appear.
4. Choose **Container settings** and set values for the segmentation fields listed in step 1. It's possible that some of the fields don't apply to the segmentation markers you choose.
5. If necessary, create the video encode in the output, then choose the video encode. In the **Codec settings** field, choose the codec. More fields appear.
6. Choose the **Frame rate** section and set the following fields:

- **Framerate control**: We recommend you choose **Specified**. The option **Initialize_from_source** doesn't work well with pipeline locking.
- **Framerate numerator** and **Framerate denominator**: Set the desired framerate for the output. Make sure that the conversion from input framerate to output framerate meets the requirements.

Troubleshooting

Pipeline locking ensures that the two pipelines in a standard channel are frame accurate with each other, in the output groups where MediaLive performs pipeline locking.

If you or the downstream system notice that the pipelines are not synchronized, perform the following troubleshooting:

- Make sure that MediaLive supports pipeline locking for the type of input in your channel.
- Verify that the timecode requirements are met:
  - Make sure that the input source has an embedded timecode.
  - If you chose epoch-locking mode, make sure that the embedded timecode is within 2 minutes of epoch time.

  If an input source has sections where there is no embedded timecode, MediaLive stops performing frame-accurate pipeline locking. MediaLive automatically falls back to performing approximate pipeline locking. Whenever the embedded timecode reappears, MediaLive resumes frame-accurate pipeline locking.
- Make sure that MediaLive supports pipeline locking in the affected output group. Pipeline locking doesn't apply to all output group types.

- Make sure that you changed the **Framerate control** so that it is *not* **Initialize_from_source**.
- Make sure that the input framerate and output framerate are a simple conversion of each other.
- If the framerate within the source changes, it's possible that MediaLive can't perform pipeline locking for the duration because for that section of video, there is no simple framerate conversion.
- Make sure that you remembered to set up segmentation markers in a UDP output group. For the other supported output groups, you don't need to worry about this because their outputs are always segmented.

- Make sure that you set up the segmentation marker type that your downstream system expects.

Implementing pipeline redundancy

You can set up a channel with two encoding pipelines, to provide resiliency within the channel processing pipeline.

When you set up the channel with two encoding pipelines, both pipelines ingest the source content and produce output. If the current pipeline fails, the downstream system can detect that it is no longer receiving content and can switch to the other output. There is no disruption to the downstream system. MediaLive restarts the second pipeline within a few minutes.

A channel that has two encoding pipelines is called a *standard channel*.

If you don't want to implement pipeline redundancy, you set up the channel as a *single-pipeline channel*. If the single pipeline fails, MediaLive stops producing output to deliver to the downstream system.

Topics

Deciding whether to implement pipeline redundancy

To determine the channel class to implement, you must decide if you want to and are able to implement pipeline redundancy.

Step 1: Decide if you want to implement pipeline redundancy

Decide if you *want* to implement pipeline redundancy. As well as the obvious benefit of redundant pipelines, consider the following points:.

- If you are sending output to AWS Elemental MediaPackage, you might want to implement pipeline redundancy in order to support *input redundancy* in MediaPackage. MediaLive will send

two identical outputs to the two inputs on the MediaPackage channel. If there is a pipeline failure in MediaLive, MediaPackage has logic to seamlessly switch the input it uses.

- Weigh the benefit of a standard channel against the difference in processing charges for a standard channel compared to a single-pipeline channel. For information about charges for channels, see https://aws.amazon.com/medialive/pricing/.
- If you decide you don't yet want to implement pipeline redundancy, you can set up to leave open the option of implementing it later on. The procedures later in this section explain how to set up in this way.

Step 2: Decide if you can implement pipeline redundancy

If you decide that you want to set up a standard channel, you must determine if you *can* set up a standard channel. Follow these steps:

- Contact the upstream system to determine if they can send you two source streams for each input. If they can't, then you can't set up as a standard channel.

  In a multiple-input channel, all the inputs must have two source streams. If you have source content coming from several upstream systems, every upstream system must be capable of providing two sources. If they can't all provide two sources, you can't set up as a standard channel.

- Contact the downstream system to determine if the downstream system can handle two sets of identical outputs from MediaLive and to switch as required. Note that, as described earlier in this decision section, MediaPackage can always handle two outputs.

  If the downstream system doesn't have this ability, there is no advantage to setting up as a standard channel.

Step 3: Follow the correct procedure

After you have identified the pipeline redundancy option that you will implement in the channel, see the following sections for more information:

- If you want to implement pipeline redundancy immediately, and the upstream system can provide two source streams, then see the section called “Standard channel”.

- If you don't want to implement pipeline redundancy for now, but you want to allow for easy upgrade to pipeline redundancy later, then see the section called “Single-pipeline channel with upgrade options”.
- If you don't want to implement pipeline redundancy now or in the future, then see the section called “Single-pipeline channel without upgrade”.

Setting up a standard channel

If you want to implement pipeline redundancy with a new channel, make sure that you set up the inputs as standard-class inputs and set up the channel as a standard channel.

Follow these guidelines when you plan the workflow:

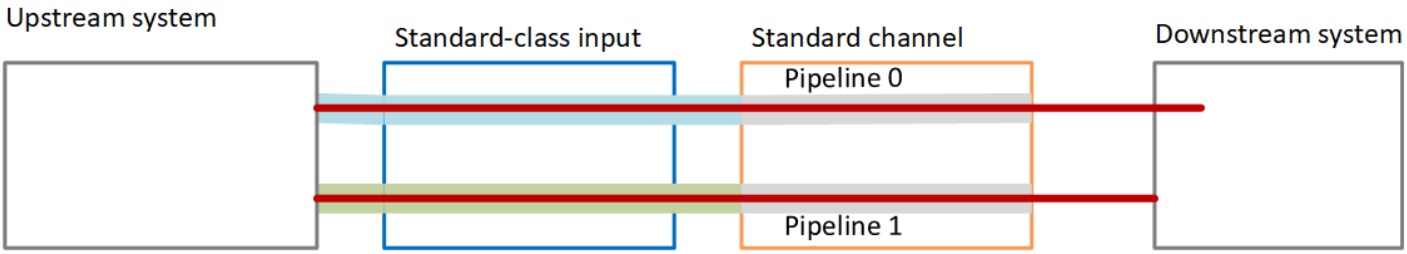
- Make sure that the upstream system can provide you with two instances of the source content. See the section called “Assess source formats and packaging”.
- When you create inputs, set up all the inputs as standard-class inputs.

  Some inputs — CDI inputs and RTP inputs — are always set up as standard-class inputs. For all other inputs, set the **Input class** field to **Standard input**.
- When you create the channel, do the following:
  - Set up the channel as a standard channel. See the section called “Channel and input details”.
  - At the step to attach inputs to the channel, attach only standard-class inputs. If you try to attach a single-class input to a standard channel, you won't be able to create the channel.
- Contact the upstream system and request that they provide two content sources.

How pipeline redundancy works

When you set up a standard channel, the channel has two pipelines—pipeline 0 and pipeline 1. Each input also contains two pipelines. A content source is connected to each pipeline.

As this diagram illustrates, the upstream system provides two instances of the content to the input. One instance goes to the pipeline that is indicated by the blue line, the other goes to the pipeline indicated by the green line. Each of these lines is attached to one of the two pipelines in the channel. The channel produces two identical instances of the output for the downstream system. The downstream system chooses to handle one instance (the output from blue pipeline) and to ignore the other instance (the output from the green pipeline).

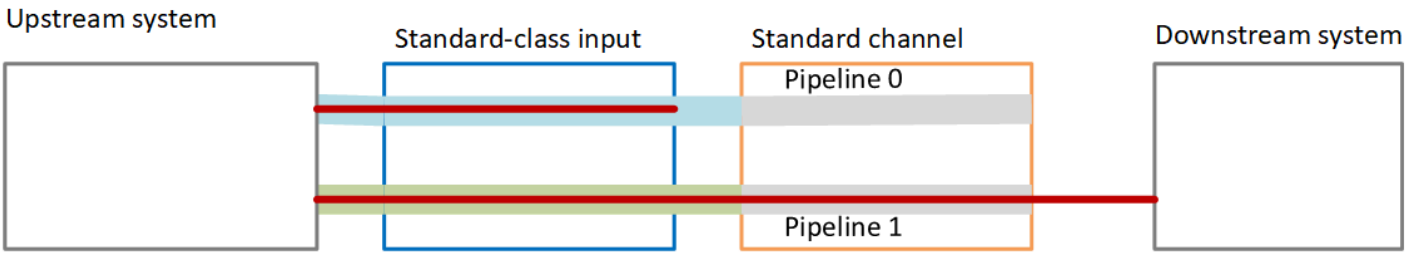


Failure handling

There might be a problem that causes a pipeline to stop functioning.

- If the failed pipeline is the pipeline that the downstream system is handling (for example, the blue pipeline), the downstream system can switch to the other output.
- After a few minutes, the failed pipeline automatically restarts and produces output. The downstream system can continue to handle the output from the green pipeline, or it can go back to the blue pipeline. That decision has no impact on MediaLive.

In this diagram, notice that the upstream system is still sending source content to the blue pipeline, which indicates that the upstream system is working but pipeline 0 has failed. The downstream system has started handling pipeline 1 instead, using the source content from the green pipeline.



Setting up a single-pipeline channel with upgrade options

When you first create the channel, you might want to set it up without pipeline redundancy. But you might want to allow for easy upgrade to pipeline redundancy later.

Follow these guidelines when you plan the workflow:

- When you create inputs, set up all the inputs as standard-class inputs.

  Some inputs — CDI inputs and RTP inputs — are always set up as standard-class inputs. For all other inputs, set the **Input class** field to **Standard input**.

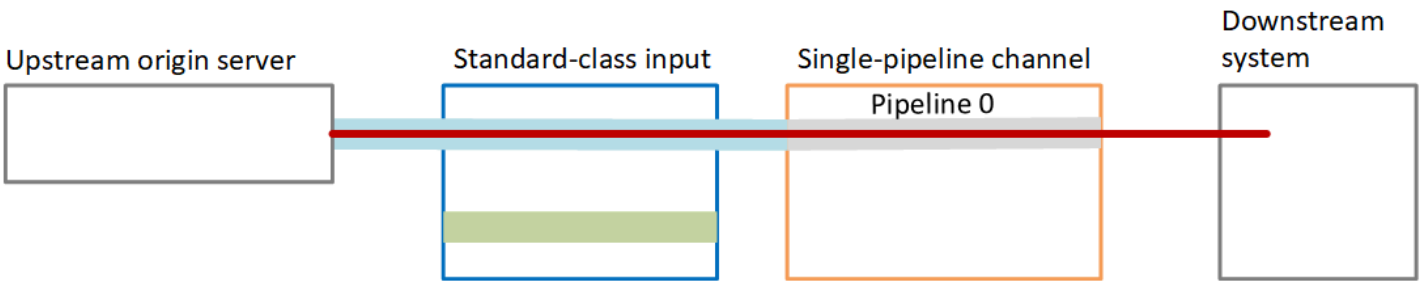
- When you create the channel, do the following:
  - Set up the channel as a single-pipeline channel. See the section called "Channel and input details".
  - At the step to attach inputs to the channel, double-check that the inputs you attach are standard-class inputs.
- Contact the upstream system and request that they provide *one* content source.

How a single-pipeline channel works

When you set up a single-pipeline channel with the option to easily upgrade, the channel is a single-pipeline channel but the inputs are all standard-class inputs.

- The channel contains one pipeline—pipeline 0.
- Each standard-class input contains two pipelines. However, only one of the pipelines is connected to a content source. The other input pipeline is inactive.

As this diagram illustrates, the upstream system provides one instance of the source content to the input, to the pipeline that is indicated by the blue line. The input provides that one instance to the one pipeline in the channel. The channel produces one instance of the output for the downstream system. The other pipeline in the input (the green pipeline) is always inactive.



Failure handling

If there is a problem that causes a pipeline to stop functioning, MediaLive stops producing output. The downstream system stops receiving output.

Setting up a single-pipeline channel without upgrade potential

If you don't want to implement pipeline redundancy now or in the future, you set up the channel as a single-pipeline channel, and you set up the inputs as single-class inputs, where possible.

> **Note**
>
> Before you decide to implement this option, read the information about setting up without pipeline redundancy, but with the option to easily upgrade later on.

Follow these guidelines when you plan the workflow:

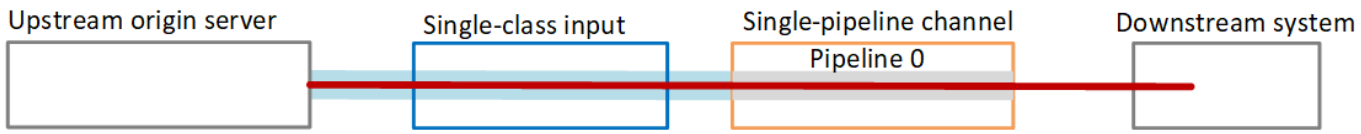
- When you create inputs, set up the inputs as follows:
  - Set up CDI inputs and RTP inputs as standard inputs, because that's the only way to set them up.
  - Set up all other inputs as single-class inputs. To set up the input in this way, set the **Input class** field to **Single input**.
- When you create the channel, do the following:
  - Set up the channel as a single-pipeline channel. See the section called "Channel and input details".
  - At the step to attach inputs to the channel, attach the inputs that you have identified. The inputs might be both standard-class inputs and single-class inputs.
- Contact the upstream system and request that they provide *one* content source. Even if your channel includes CDI inputs or RTP inputs, the upstream system for those inputs should provide only one source.

How a single-pipeline channel works

When you set up a single-pipeline channel without any upgrade provision, the channel is a single-pipeline channel. The inputs can be a combination of single-class inputs and standard-class inputs.

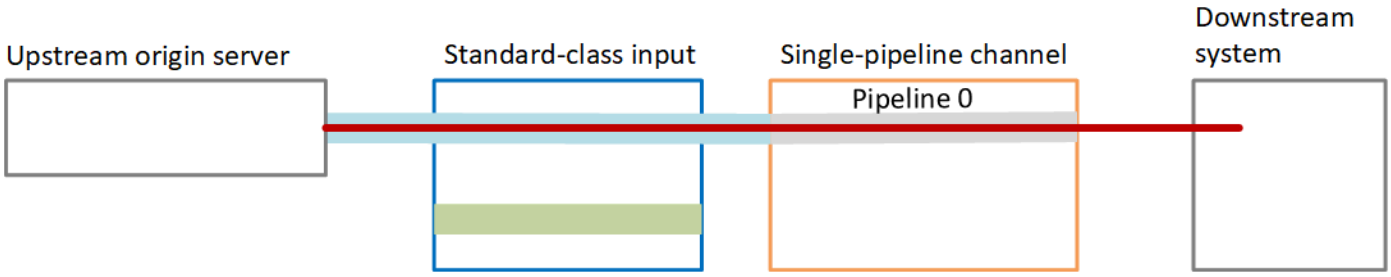
- The channel contains one pipeline—pipeline 0.
- Each single-class input that is attached to the channel contains one pipeline. The input is connected to one content source.

  As this diagram illustrates, the upstream system provides one instance of the source content to the input, to the pipeline that is indicated by the blue line. The input provides that one instance to the one pipeline in the channel. The channel produces one instance of the output for the downstream system.



- Each CDI input or RTP input that is attached to the channel contain two pipelines. However, only one of the pipelines is connected to a content source. The other input pipeline is inactive.

  As this diagram illustrates, the upstream system provides one instance of the source content to the input, to the pipeline that is indicated by the blue line. The input provides that one instance to the one pipeline in the channel. The channel produces one instance of the output for the downstream system. The other pipeline in the input (the green pipeline) is always inactive.



Failure handling

If there is a problem that causes a pipeline to stop functioning, MediaLive stops producing output. The downstream system stops receiving output.

Changing pipeline redundancy in an existing channel

To enable or disable pipeline redundancy on an existing channel, you must update the channel class.

Changing the channel to a single-pipeline channel

You can change a standard channel to single-pipeline, to remove one of the pipelines in the channel and to remove pipeline redundancy.

To change the channel class, the channel must be idle (not running).

To change the channel class to a single-pipeline channel

1. On the **Channels** page, choose the channel. (Don't choose the channel name.)

2. On the menu, choose **Actions**, then choose **Other channel actions**, then choose **Update channel class to SINGLE_PIPELINE**.
3. In the dialog box, choose **Confirm**. MediaLive performs the following actions:

   - It removes the second pipeline (pipeline 1) in the channel.
   - It removes the second destination address in each output group.
   - It *doesn't* remove the second endpoint on the inputs. The inputs aren't changed in any way. Instead, when you restart the channel, MediaLive simply ignores the second endpoint.

   While MediaLive is performing these actions, the channel has a status of **UPDATING**. When the update is completed, the status changes to **IDLE**.
4. You might want to notify the upstream system for each push input that it no longer needs to push input to the second endpoint. You also might want to notify the downstream system for each output group that it should no longer expect output at its second destination.

Changing the channel class to standard – option A

You can change a single-pipeline channel to a standard channel. Follow this procedure if you originally set up the single-pipeline channel with standard-class inputs and upgrade potential.

Perform these steps:

- Arrange with your upstream systems to start sending two instances of the source content.
- Stop the channel.
- Change the channel class to standard class. See the steps after this list.

  You have now upgraded the channel from a single-pipeline channel with standard-class inputs to a standard channel with standard-class inputs.
- Restart the channel.

To change the channel class

1. Obtain a second destination address for each output group. Each address is at the downstream system of each output group.

   For example, if the channel has an HLS output group (with an HTTPS server as its downstream system) and an Archive output group (with an Amazon S3 bucket as its downstream system),

you must enter the URL to a new destination address at the HTTPS server, and the URL to a new folder in the Amazon S3 bucket.

Plan these destinations now, in the same way as you planned the destination addresses when you originally set up the channel. You might need to contact the owner of each downstream system.

2. On the **Channels** page, choose the channel. (Don't choose the channel name.)
3. On the menu, choose **Actions**, **Other channel actions**, **Update channel class to STANDARD**.
4. In the dialog box, choose **Confirm**.
5. On the **Update channel class to Standard** page, enter the destination addresses that you identified in step 1. There is one field for each output group in the channel.
6. Choose **Submit**. MediaLive updates the channel and creates a new pipeline called pipeline 1. The source for this pipeline is the previously dormant URL. When you start the channel, MediaLive ingests content from that URL, produces output, and sends the output to the new destinations in every output group.

Changing the class—option B

You can change a single-pipeline channel to a standard channel. Follow this procedure if you originally set up a single-pipeline channel with single-class inputs.

Perform these steps:

- Arrange with your upstream systems to start sending two instances of the source content.

- Stop the channel.
- Detach each single-class input. To detach the inputs, you must edit the channel and remove the attached inputs.
- Edit each input to convert it to standard-class, and to add a second source.
- Edit the channel to change the channel class to standard class. See the steps after this list.
- Edit the channel to reattach each input.

  You have now upgraded the channel from a single-pipeline channel with single-class inputs to a standard channel with standard-class inputs.
- Restart the channel.

To change the channel class

1. Obtain a second destination address for each output group. Each address is at the downstream systems of each output group.

   For example, if the channel has an HLS output group (with an HTTPS server as its downstream system) and an archive output group (with an Amazon S3 bucket as its downstream system), you must enter the URL to a new destination address at the HTTPS server, and the URL to a new folder in the Amazon S3 bucket.

   Plan these destinations now, in the same way as you planned the destination addresses when you originally set up the channel. You might need to contact the owner of each downstream system.
2. Edit the URLs in every single-class input to include a second URL, for the second source that will provide content to the newly added pipeline.

   - For a push input, edit the input to include an address for the second input source. Give that address to the owner of the upstream system, so that they can push source content to that address. You should also find out from the upstream system the address that the new source will be pushed from. Make sure that this address is covered by the input security group for the channel.

   - For a pull input, obtain a new address from the owner of the downstream system. Edit the input to include that address. After the second pipeline is created, MediaLive will be able to pull the second source content (for the second pipeline).
3. On the **Channels** page, choose the channel. (Don't choose the channel name.)
4. On the menu, choose **Actions**, **Other channel actions**, **Update channel class to STANDARD**.
5. In the dialog box, choose **Confirm**.
6. On the **Update channel class to STANDARD** page, enter the destination addresses that you identified in step 1. There is one field for each output group in the channel.
7. Choose **Submit**. MediaLive updates the channel and creates a new pipeline called pipeline 1. When you start the channel, MediaLive sends the output from this pipeline to the new destinations in every output group.

Implementing resiliency in the channel

AWS Elemental MediaLive has several features that provide resiliency in the channel:

- Automatic input failover – You can set up two inputs in an *input failover pair*. Setting up this way provides resiliency in case of a failure either in the upstream system, or between the upstream system and the channel. For more information, see the section called “Automatic input failover”.
- Input loss behavior – You can set up the channel to control how MediaLive behaves when video input is lost. This feature covers all inputs—those that are set up with automatic input failover, and those that aren't.

  For more information, see the section called “Input loss handling”.
- Pipeline redundancy – You can set up the channel with two pipelines, to provide resiliency within the channel pipeline. This feature is controlled by the channel class feature of the channel. For more information, see the section called “Pipeline redundancy”.

SCTE-35 message processing

You can configure an AWS Elemental MediaLive channel to handle SCTE-35 messages and SCTE-104 messages. These messages provide information about ad avails (advertisement availability events), and other non-ad avail events, and other non-ad avail events (such as programs and chapters).

Topics

About message processing

SCTE-35 messages are messages that can be included in a source MPEG-2 transport stream (TS). SCTE-104 messages are messages that can be included in source content from an AWS Elemental Link hardware device. SCTE-104 messages are automatically converted into SCTE-35 messages as soon as MediaLive ingests the input.

> **Note**
>
> To use the ad avail features of MediaLive, you should be familiar with the SCTE-35 standard and optionally with the SCTE-67 standard. You should also be familiar with how the input that you are encoding implements those standards.
> This section assumes that you are familiar with creating or editing a channel, as described in *Setup: Creating a channel* .

Support for SCTE-35 on the input side

On the input side, SCTE-35 messages can only appear in inputs containing MPEG-2 transport streams (TS). You can set up a channel so that if an input includes these messages, the messages are either processed during ingest (passed through) or ignored.

Support for SCTE-35 on the output side

On the output side, if you set up to pass through the input (rather than remove it), then you can set up each output so that the SCTE-35 messages from the input are turned into cueing information that is appropriate for that output type. This cueing information can be in the form of one or both of the following:

- SCTE-35 messages in a TS output
- Manifest (or sparse track) decoration

You set up each output separately, so that you can set up some outputs to include cueing information and some to exclude it.

As an adjunct to the ad avail information, you can also set up the outputs to blank out the video, audio, and captions within the cueing information.

Topics

Supported features by input type

SCTE 35 messages can appear only in the following types of MediaLive inputs:

- Elemental Link inputs
- HLS inputs
- MediaConnect inputs
- RTP inputs
- Transport Stream (TS) File inputs
- AWS CDI inputs

The following table shows which inputs might include ad avail information and how MediaLive handles that information. To read the table, find an input in the first column, then read across in the row.

Input	Interpret SCTE-35 messages in the transport stream	Interpret ad avail information in the input manifest		
Elemental Link	Yes	Not applicable		
HLS	Yes	Yes		
MediaConnect	Yes	Not applicable		
RTMP	No	Not applicable		
RTP	Yes	Not applicable		

Input	Interpret SCTE-35 messages in the transport stream	Interpret ad avail information in the input manifest
Transport Stream (TS) file	Yes	Not applicable
AWS CDI	Yes	Not applicable

Supported output features

Manifest decoration

You can set up an output so that its manifest is decorated with ad avail information. Manifest decoration works on two sources of ad avail information:

- Ad avail information found in the channel input, if the input is a transport stream (TS)
- Ad avail information from SCTE-35 messages added to the output using the MediaLive schedule

Manifest decoration applies only to HLS outputs, MediaPackage outputs, and Microsoft Smooth outputs:

- You can set up HLS outputs so that their manifests are decorated according to one of the following styles:
  - Adobe
  - Elemental
  - SCTE-35 enhanced
- MediaPackage outputs are always set up so that their manifests are decorated. The marker style is always SCTE-35 enhanced style. Keep in mind that if you don't actually want SCTE-35 messages in the output that you deliver from AWS Elemental MediaPackage, then on the AWS Elemental MediaPackage side you can set up the channel to remove the markers.
- You can set up Microsoft Smooth outputs so that the sparse track includes instructions that correspond to the original SCTE-35 message content.

You must set up the channel for the behavior that you want. For more information, see the section called " Manifest decoration".

Blanking and blackout

The *cue out* and *cue in* instructions in SCTE-35 messages in TS inputs line up with specific content in the video, audio, and captions streams. You can set up so that this content is blanked out in the output:

- To blank out content for ad avails, use the ad avail blanking feature.
- To blank out content for other messages, use the blackout feature.

You must set up the channel for the behavior that you want.

For more information, see the section called "Ad avail blanking" and the section called "Blackout".

SCTE-35 passthrough

You can set up TS outputs so that all the SCTE-35 messages from the input are passed through to the output. Or you can set up to remove these messages from the output.

The behavior that you want must be set up in the channel. For more information, see the section called "Passthrough of SCTE 35 messages".

Inserting SCTE-35 messages using the schedule

You can insert SCTE-35 messages in TS outputs using the channel schedule. For example, you can add an action in the channel schedule to insert a splice insert in the running channel.

The main use case for this feature is to add SCTE-35 messages to the output, when the input doesn't already include SCTE-35 messages.

For more information, see the section called "Inserting messages".

Processing features – default behavior

The default handling of SCTE-35 by MediaLive is the following:

- No passthrough – Remove SCTE-35 messages in any data stream outputs. There is one exception: for MediaPackage outputs, passthrough is always enabled.

- No blanking or blackout – Do not blank out video content for any events. Leave the content as is.
- No manifest decoration – Do not convert any SCTE-35 messages to event information in any output manifests or data streams. There is one exception: for MediaPackage outputs, manifest decoration is always enabled and can't be disabled.

If this is the behavior that you want, you don't need to read any further in this SCTE-35 section.

Typically, you change these defaults only if you want to include ad avail information in the channel outputs. The following are examples of when you change the defaults

- You enable passthrough.
- You enable manifest decoration, if your channel includes HLS, MediaPackage, or Microsoft Smooth output groups.
- You blank or blackout video content depending on your agreement with the content provider.

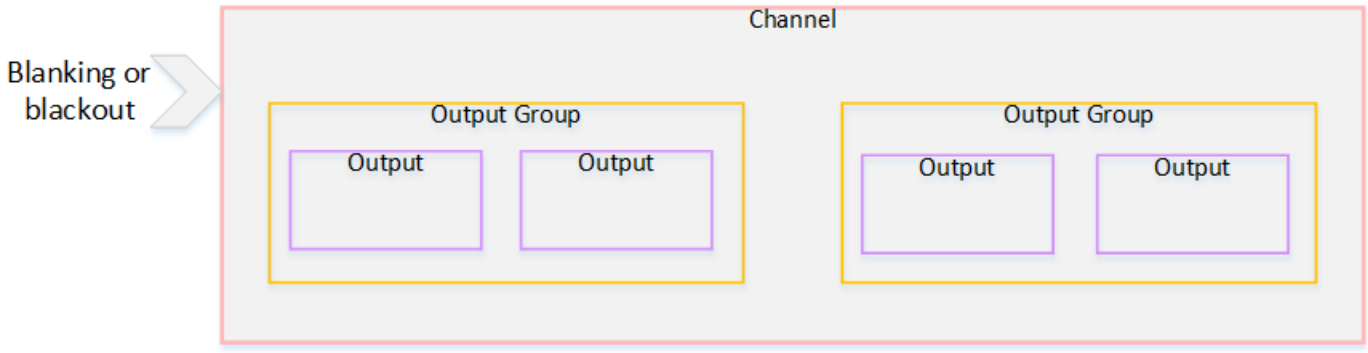
Scope of processing by feature

The SCTE-35 features have different scopes in terms of the output groups and outputs that they affect:

Blackout or ad avail blanking

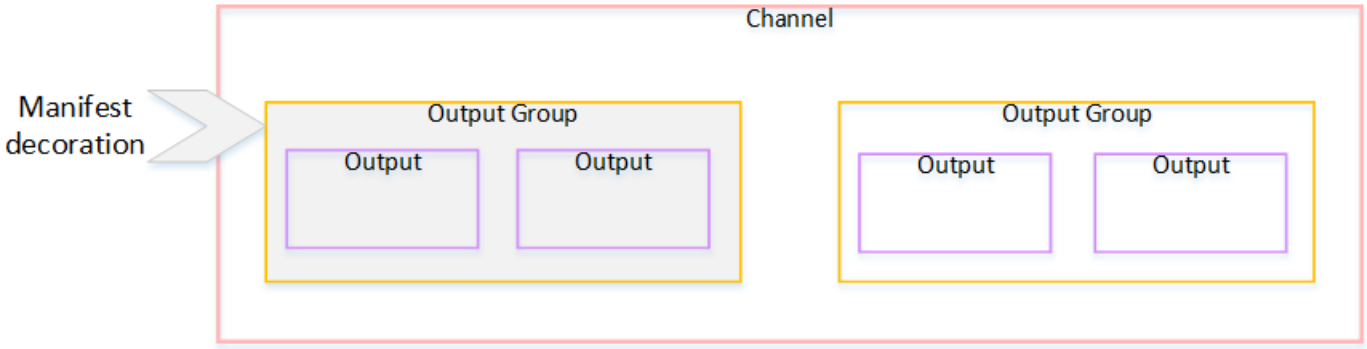
Blackout applies at the *global output* level. If you enable blackout, all the relevant content in every output in every output group is blanked.

Ad avail blanking also applies at the *global output* level. If you enable blanking, all the ad avails in every output in every output group are blanked.



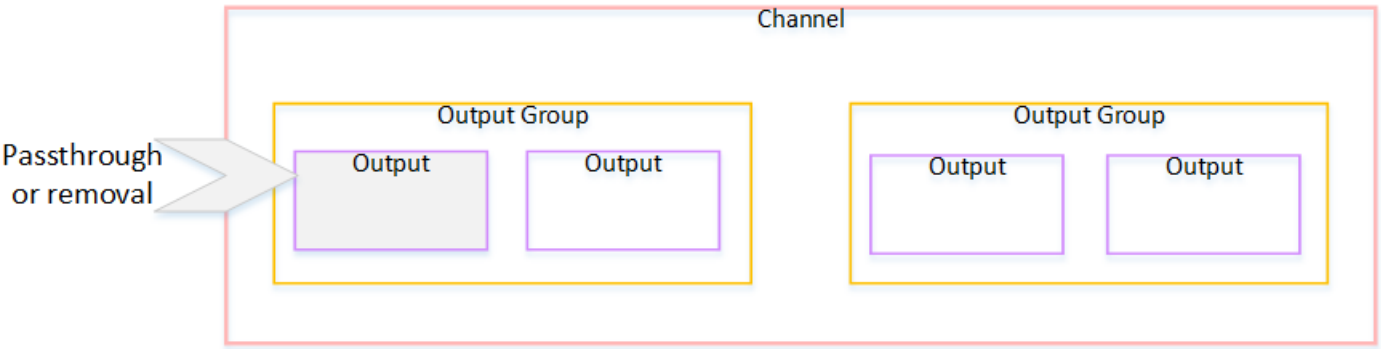
Decoration

Manifest decoration applies at the *output group* level. If you enable manifest decoration in an output group, all the outputs in that output group have their manifests decorated.



SCTE-35 passthrough or removal

SCTE-35 passthrough or removal applies at the *output* level. You can enable passthrough or removal in individual TS outputs. The messages are passed through or removed only in those outputs.



Supported features by output type

This section describes which SCTE-35 features apply to the various types of output.

Topics

- Archive output with MPEG-2 container
- Frame capture output
- HLS output
- MediaPackage output
- Microsoft Smooth output
- RTMP output
- UDP output

Archive output with MPEG-2 container

In an Archive output (a transport stream in an MPEG-2 container), MediaLive supports SCTE-35 features as follows:

- Passthrough of the SCTE-35 messages – Supported.
- Manifest decoration – Not supported because these outputs don't have manifests.
- Blanking and blackout – Applicable. Content in the output is blanked or blacked out if the features are enabled at the channel level.

Be careful of setting up so that you have removed messages from the input (passthrough disabled) and you have not enabled blanking and blackout. In this case, the video content that was marked by messages (in the input) will not be marked (in the output).

- If you have the rights to that video content, there is no problem setting up this way.
- If you don't have the rights, then the only way to find that content will be to look for the IDR i-frames that identify where the SCTE-35 message used to be.

Frame capture output

In a frame capture output, MediaLive supports SCTE-35 features as follows:

- Passthrough of the SCTE-35 messages – Not applicable.
- Manifest decoration – Not supported because these outputs don't have manifests.
- Blanking and blackout – Applicable. Content in the output is blanked or blacked out if the features are enabled at the channel level.

A frame capture output doesn't support passthrough of the SCTE-35 messages. However, if blanking or blackout is enabled (at the channel level), then content that falls between the start and stop of the blackout is blanked or blacked out, even though no SCTE-35 messages are present.

HLS output

In an HLS output (a transport stream), MediaLive supports SCTE-35 features as follows:

- Passthrough of the SCTE-35 messages – Supported.
- Manifest decoration – Supported.

- Blanking and blackout – Applicable. Content in the output is blanked or blacked out if the features are enabled at the channel level.

MediaLive supports the following combinations of passthrough and manifest decoration:

- Passthrough enabled, decoration enabled.
- Passthrough disabled, decoration enabled.
- Passthrough disabled, decoration disabled. Be careful of setting up with this combination but leaving blanking and blackout disabled. In this case, the video content that was marked by messages (in the input) are not marked (in the output). In addition, the manifests don't have information for identifying that video content.
  - If you have the rights to that video content, there is no problem setting up this way.
  - If you don't have the rights, the only way to find that content is to look for the IDR i-frames that identify where the SCTE-35 message used to be.

MediaPackage output

In a MediaPackage output, MediaLive supports SCTE-35 features as follows:

- Passthrough of the SCTE-35 messages – Always enabled.
- Manifest decoration – Always enabled.
- Blanking and blackout – Applicable. Content in the output is blanked or blacked out if the features are enabled at the channel level.

Microsoft Smooth output

In a Microsoft Smooth output, MediaLive supports SCTE-35 features as follows:

- Passthrough of the SCTE-35 messages – Not applicable. SCTE-35 messages are never included in this output.
- Manifest decoration – Not supported because these outputs don't have manifests. However, you can set up to include instructions in the sparse track.
- Blanking and blackout – Applicable. Content in the output is blanked or blacked out if the features are enabled at the channel level.

Be careful of setting up so that you have the following combination:

- You have not enabled sparse track.
- You have not enabled blanking and blackout.

In this case, the video content that was marked by messages (in the input) is not marked (in the output).

- If you have the rights to that video content, there is no problem setting up this way.
- If you don't have the rights, it is impossible to find these blanks and blackouts programmatically in a Microsoft Smooth output.

RTMP output

In an RTMP output, MediaLive supports SCTE-35 features as follows:

- Passthrough of the SCTE-35 messages – Not applicable.
- Manifest decoration – Not supported.
- Blanking and blackout – Applicable. Content in the output is blanked or blacked out if the features are enabled at the channel level.

UDP output

In a UDP output (a transport stream), MediaLive supports SCTE-35 features as follows:

- Passthrough of the SCTE-35 messages – Supported.
- Manifest decoration – Not supported because these outputs don't have manifests.
- Blanking and blackout – Supported.

Be careful of setting up so that you have removed messages from the input (passthrough disabled) and you have not enabled blanking and blackout. In this case, the video content that was marked by messages (in the input) is not marked (in the output).

- If you have the rights to that video content, there is no problem setting up this way.
- If you don't have the rights, then the only way to find that content is to look for the IDR i-frames that identify where the SCTE-35 message used to be.

Getting ready: Set the SCTE-35 source—segments or manifest

If you have HLS inputs in the channel, you must configure the input to identify the source of the SCTE-35 messages. There are two possible sources:

- The segments in the transport stream (TS). This type of source applies to all inputs that can include SCTE-35 messages. Unless a specific SCTE-35 packet identifier (PID) is selected, the first PID present in the TS will be used.
- Tags in an HLS input manifest. This type of source applies only to HLS inputs.

To set the source in a non-HLS input

1. On the **Create/Edit channel** page, in the navigation pane, choose **Input attachments**.
2. In **General input settings**, complete the following field:
   - **SCTE-35 PID**: Enter the PID value. If the value is left blank, the first SCTE-35 PID present in the input will be selected.
3. If appropriate, repeat for other **Input attachments**.

> **Note**
>
> You only need to follow this next procedure for HLS inputs. For all other inputs, the source of the SCTE-35 messages is always the TS segments.

To set the source in an HLS input

1. On the **Create/Edit channel** page, in the navigation pane, choose **Input attachments**.
2. For each HLS input, in **Network input settings**, in **HLS input settings**, choose **HLS input**. More fields appear.
3. Set SCTE-35 source to **SEGMENTS** (the default) or **MANIFEST**.

Topics

- How MediaLive inserts the message: preroll

Supported manifest formats

Read the following sections if you set up to use the HLS input manifest as the SCTE-35 source.

MediaLive can generate SCTE35 splice insert messages from **EXT-X-CUE-OUT** and optionally **EXT-X-CUE-IN** tags within the source HLS manifest. Following are examples of supported formats for these tags.

- `#EXT-X-CUE-OUT:DURATION=60.000`
- `#EXT-X-CUE-OUT:DURATION="60.000"`
- `#EXT-X-CUE-OUT:60.000`
- `#EXT-X-CUE-OUT:"60.000"`
- `#EXT-X-CUE-IN`

How MediaLive creates the SCTE-35 messages

For each `EXT-X-CUE-OUT`, MediaLive creates an SCTE-35 message of type splice insert with the following data:

- `splice_event_id`: A number that increments, starting with 1 for the first CUE-OUT message that MediaLive creates from the current input.
- `out_of_network_indicator`: true (1)
- `program_splice_flag`: true (1)
- `duration_flag`: true (1)
- `break_duration`:
  - `auto_return`: 1
  - `reserved`: 0
  - `duration`: The duration from the manifest, converted to 90kHz ticks. For example, 15 seconds is 1350000 ticks.
- `splice_immediate_flag`: 0 (false)
- `splice_time`: Use the video PTS of the first frame of the video segment that follows this EXT-X-CUE-OUT in the input manifest

- unique_program_id: 0
- avail_num: A number that increments, starting with 1 for the first CUE-OUT message that MediaLive creates from the current input.
- avails expected: 0

For each EXT-X-CUE-IN, MediaLive creates an SCTE-35 message of type splice insert with the following data:

- splice_event_id: The ID from the most recent EXT-X-CUE-OUT in the manifest.
- out_of_network_indicator: false (0)
- program_splice_flag: true (1)
- duration_flag: false (0)
- splice_immediate_flag: 0 (false)
- splice_time: Use the video PTS of the first frame of the video segment that follows this EXT-X-CUE-IN in the input manifest
- unique_program_id: 0
- avail_num: The value from the most recent EXT-X-CUE-OUT
- avails expected: 0

How MediaLive inserts the message: preroll

MediaLive includes a preroll when it inserts the SCTE-35 message that corresponds to the CUE-OUT. This preroll is 5 seconds in advance of the splice_time in the SCTE-35 message.

MediaLive reduces the preroll if the channel doesn't have enough buffering to allow the preroll. The buffer, in seconds, is the product of the following:

- Input segment duration, which is specified in the input manifest
- Number of segments to include in the buffer. You set this value in the **Buffer segments** field when you attach the HLS input.

For example, if the segment duration is 6 seconds and the number of segments is 3, then the buffer is 18 seconds.

Ensuring an adequate preroll

If the calculated buffer for your input is shorter than 5 seconds, MediaLive reduces the preroll. MediaLive might reduce the preroll to 0, which would mean that the PTS value of the SCTE35 message equals the PTS of the splice time.

To avoid an inadequate preroll, we recommend that you make sure that the buffer is *at least* equal to the preroll, plus one segment. Follow these steps:

- Step 1: Calculate the minimum buffer, in seconds, for your input: Preroll in seconds + length of one segment in seconds
- Step 2: Calculate the number of segments in that minimum buffer: Divide the minimum buffer by the segment length
- Step 3: Round that minimum up to a whole number. Or that minimum is less than 3, round that number up to 3.
- Step 4: Enter this number (or a bigger number, if you want) in the **Buffer segments** in the Input attachment.

For example, assume the segment length is 2 sec.

- Step 1: 5 + 2 = 7
- Step 2: 7 secs divided by 2 = 3.5
- Step 3: Round up to 4.
- Step 4: Enter that number (or a bigger number) in the **Buffer segments** in the Input attachment.

Getting ready: Set the ad avail mode

You must set the mode for SCTE 35 handling. The blanking, blackout, and manifest decoration features of MediaLive work different depending on the mode.

To set the ad avail mode

1. In the channel that you are creating, in the navigation pane, choose **General settings**. Choose **Avail configuration**.
2. Complete the fields as follows:

Field	Description
Avail settings	If your organization has a POIS server that handles decisions about SCTE 35 messages, then choose **ESAM** and read the section called "POIS signal conditioning" now. Otherwise, choose **SCTE-35 splice insert** or **SCTE-35 time signal apos**. The mode to choose depends on the message types that you expect to be present in the source, and on how you want to handle those messages. See the table later on this page.
Ad avail offset	Set a value, if desired. For details about a field on the MediaLive console, choose the **Info** link next to the field.
web_delivery_allowed_flag	Typically, leave as **Follow**. For information about these fields, see the section called "Ad avail blanking restriction flags".
no_regional_blackout_flag	

Field	Description
SCTE-35 segmentation scope	This field affects segment breaks in the video output encodes, in output groups that contain transport streams. In other words, in HLS, MediaPackage, Multiplex, and UDP output groups. This field doesn't affect output groups that don't contain transport streams. The field controls how segmentation in these TS output groups is affected by SCTE 35 messages. The field is particularly important if some of these TS output groups have SCTE 35 passthrough enabled (they are *SCTE 35-enabled output groups*), and some don't have it enabled. Choose the desired appropriate value: • **ALL_OUTPUT_GROUPS**: MediaLive inserts the SCTE 35-triggered segment break in all output groups. In non-SCTE 35-enabled output groups, this behavior might result in unnecessary segment breaks or in inconsistent segment break lengths. • **SCTE35_ENABLED_OUTPUT_GROUPS**: MediaLive inserts the SCTE 35-triggered segment break only in *SCTE 35-enabled output groups*. This is the recommended value, because it reduces unnecessary segment breaks in output groups that aren't SCTE 35 enabled.

This table identifies how the two different ad avail modes work. It identifies the combinations of message type and segmentation type that each mode considers as an *ad avail*. Note that in both modes, MediaLive looks at both splice insert messages and time signal messages.

To read this table, find a message type in the first column and a segmentation type in the second column. The third and fourth columns specify whether MediaLive treats this message combination as an ad avail when the mode is splice insert mode and when the mode is timesignal APOS mode.

Message type ID	Segmentation type and IDs	Does splice insert mode treat this message as an ad avail	Does timesignal APOS mode treat this message as an ad avail
splice insert	No segmentation descriptor present	No	No
	Provider advertise ment (0x30/0x31)	Yes, it treats it as an ad avail	No
	Distributor advertise ment (0x32/0x33)	Yes, it treats it as an ad avail	No
	Provider placement opportunity (0x34/0x35)	Yes, it treats it as an ad avail	No
	Distributor placement opportunity (0x36/0x37)	Yes, it treats it as an ad avail	No
	Break (0x22/0x23)	Yes, it treats it as an ad avail	No
	Other: Programs, Chapters, Network, Unscheduled	Yes, it treats it as an ad avail	No
time signal	No segmentation descriptor present	Not applicable to time signal messages	Not applicable to time signal messages

Message type ID	Segmentation type and IDs	Does splice insert mode treat this message as an ad avail	Does timesignal APOS mode treat this message as an ad avail
	Provider advertise ment (0x30/0x31)	Yes, it treats it as an ad avail	No
	Distributor advertise ment (0x32/0x33)	Yes, it treats it as an ad avail	No
	Provider placement opportunity (0x34/0x35)	Yes, it treats it as an ad avail	Yes, it treats it as an ad avail
	Distributor placement opportunity (0x36/0x37)	Yes, it treats it as an ad avail	Yes, it treats it as an ad avail
	Break (0x22/0x23)	Yes, it treats it as an ad avail	Yes, it treats it as an ad avail
	Other: Programs, Chapters, Network, Unscheduled	No	No

Enabling manifest decoration in the output

You can choose to interpret SCTE-35 messages from the original input and insert corresponding instructions into the output manifest for the following outputs:

- HLS
- Microsoft Smooth (the instructions are inserted in the sparse track).

MediaPackage outputs, which are a type of HLS output, are set up with manifest decoration enabled. You can't disable decoration in these outputs.

Manifest decoration is enabled at the output group level. If you enable the feature in a specific output group, all the outputs in that group have their manifests decorated.

To include manifest decoration in some outputs and not others, you must create two output groups of the specified type, for example, two HLS output groups.

Topics

Enabling decoration – HLS

Manifest decoration is enabled at the output group level, which means that the manifests for all outputs in that group include instructions based on the SCTE-35 content.

To enable decoration

1. In the channel that you are creating, make sure that you have set the ad avail mode. See the section called “Get ready: Set ad avail mode”.
2. In the navigation pane, find the desired HLS output group.
3. In **Ad Marker**, choose **Add ad markers**.
4. For **HLS ad markers**, select the type of ad marker. For information about the different types of markers, see Sample manifests - HLS.
5. Repeat to add more types of markers, as desired.

The manifest for each output will include a separate set of tags for each type that you select.

Enabling decoration – Microsoft Smooth

With Microsoft Smooth, if you enable manifest decoration, instructions are inserted in the sparse track.

Manifest decoration is enabled at the output group level, which means that the sparse tracks for all outputs in that group will include instructions based on the SCTE-35 content.

To enable decoration

1. In the channel that you are creating, make sure that you have set the ad avail mode. See the section called “Get ready: Set ad avail mode”.
2. In the navigation pane, find the desired Microsoft Smooth output group.
3. For **Sparse track**, for **Sparse track type**, choose **SCTE_35**.
4. Complete **Acquisition point ID**, only if encryption is enabled on the output. Enter the address of the certificate.

How SCTE-35 events are handled in manifests and sparse tracks

When manifest decoration or sparse track is enabled, MediaLive inserts up to three types of information. The triggers for inserting this information depend on the mode.

Types of information

Type of instruction	When inserted
Base64	Information about all SCTE-35 messages in the output is incorporated into the manifest; the entire SCTE-35 message is added in base64 format.
Cue-out, cue-in	SCTE-35 messages that are ad avails result in the insertion of cue-out, cue-in instructions.
Blackout	Only applies to the SCTE-35 Enhanced ad marker style (for HLS output; see the section called “Enabling decoration – HLS”). SCTE-35 messages that are *not* ad avails result in the insertion of blackout start/end instructi ons, assuming that blackout is enabled. If blackout is not enabled, these instructions are not inserted.

Splice insert mode

This table describes MediaLive handling when splice insert mode is enabled. The table shows how MediaLive will react when it encounters a specific message type and segmentation type in the source.

To read this table, find a message type in the first column and a segmentation type in the second column. Then read across in the other three columns. A *Yes* indicates that MediaLive will insert this type of information in the manifest when it encounters this message type and segmentation type.

Message type ID	Segmentation type	Inserts base64 information	Inserts cue-out, cue-in information	Inserts blackout information
splice insert	No segmentation descriptor present	Yes		
	Provider advertisement	Yes	Yes	
	Distributor advertisement	Yes	Yes	
	Placement opportunity	Yes	Yes	
	Break	Yes	Yes	
	Other: Programs, Chapters, Network, Unscheduled	Yes	Yes	Yes
time signal	No segmentation descriptor present	Not applicable to time signal messages		

Message type ID	Segmentation type	Inserts base64 information	Inserts cue-out, cue-in information	Inserts blackout information
	Provider advertisement	Yes	Yes	
	Distributor advertisement	Yes	Yes	
	Placement opportunity	Yes	Yes	
	Break	Yes	Yes	
	Other: Programs, Chapters, Network, Unscheduled	Yes	No	Yes

Timesignal APOS mode

This table describes MediaLive handling when timesignal APOS mode is enabled. The table shows how MediaLive will react when it encounters a specific message type and segmentation type in the source.

To read this table, find a message type in the first column and a segmentation type in the second column. Then read across in the other three columns. A *Yes* indicates that MediaLive will insert this type of information in the manifest when it encounters this message type and segmentation type.

Message type ID	Segmentation type	Inserts base64 information	Inserts cue-out, cue-in information	Inserts blackout information
splice insert	No segmentation descriptor present	Yes		

Message type ID	Segmentation type	Inserts base64 information	Inserts cue-out, cue-in information	Inserts blackout information
	Provider advertisement	Yes		
	Distributor advertisement	Yes		
	Placement opportunity	Yes		
	Break	Yes		
	Other: Programs, Chapters, Network, Unscheduled	Yes		
time signal	Provider advertisement	Yes		
	Distributor advertisement	Yes		
	Placement opportunity	Yes	Yes	
	Break	Yes	Yes	
	Other: Programs, Chapters, Network, Unscheduled	Yes		Yes

Sample manifests - HLS

MediaLive supports the following HLS manifest styles for outputs:

- Adobe
- Elemental
- SCTE-35 Enhanced

This section describes the ad marker tagging for each style of output manifest.

> **Note**
>
> MediaLive doesn't interpret the ad avail decoration information in the manifest attached to the input source.

Ad marker: Adobe

Inserts a CUE: DURATION for each ad avail. Does not insert any CUE-OUT CONT (continuation tags) to indicate to a client player joining midbreak that there is a current avail. This does not insert a CUE-IN tag at the end of the avail.

Structure

Segment	Tag	Tag Count
Segment in which the ad avail starts.	1 CUE: DURATION tag	1

Tag contents

- CUE:DURATION contains the following:
  - duration – Duration in fractional seconds
  - id – An identifier, unique among all ad avails CUE tags
  - type – SpliceOut
  - time – The PTS time for the ad avail, in fractional seconds

Example

This is the tag for an ad avail lasting 414.171 PTS:

```
#EXT-X-CUE:DURATION="201.467",ID="0",TYPE="SpliceOut",TIME="414.171"
```

Ad marker: Elemental

Structure

Segment	Tag	Tag Count
Segment in which the ad avail starts.	CUE-OUT	1
Each succeeding segment.	CUE-OUT-CONT	0-n
Segment in which ad avail ends.	CUE-IN	1

Tag contents

- CUE-OUT contains DURATION
- CUE-OUT-CONT contains Elapsed time and Duration
- CUE-IN has no content

Example

```
#EXT-X-CUE-OUT:30.000
.
.
.
# EXT-X-CUE-OUT-CONT: 8.308/30
.
.
.
# EXT-X-CUE-OUT-CONT: 20.391/30
.
.
```

```
.
# EXT-X-CUE-IN
```

Ad marker: SCTE-35 enhanced

Structure

Segment	Tag	Tag Count
Segment in which the ad avail starts.	OATCLS-SCTE35	1
Segment in which the ad avail starts.	ASSET	1
Segment in which the ad avail starts.	CUE-OUT	1
Each succeeding segment.	CUE-OUT-CONT	0-n
Segment in which ad avail ends.	CUE-IN	1

Tag contents

- OATCLS-SCTE35 containing the base64 encoded raw bytes of the original SCTE-35 ad avail message.
- ASSET containing the CAID or UPID as specified in the original SCTE35 message.
- 1 CUE-OUT per ad avail.
- CUE-OUT-CONT containing the following:
  - The elapsed time of the avail.
  - The duration declared in the original SCTE35 message.
  - SCTE35 containing the base64 encoded raw bytes of the original SCTE-35 ad avail message.

  These lines repeat until the ad avail ends.
- CUE-IN to indicate the end of the avail.

Example

```
#EXT-OATCLS-SCTE35:/DA0AAAAAAAAAAAABQb+ADAQ6QAeAhxDVUVJQAAAO3/PAAEUrEoICAAAAAAg
+2UBNAAANvrtoQ==
#EXT-X-ASSET:CAID=0x0000000020FB6501
#EXT-X-CUE-OUT:201.467
.
.
.
#EXT-X-CUE-OUT-CONT:ElapsedTime=5.939,Duration=201.467,SCTE35=/DA0AAAA+…AAg
+2UBNAAANvrtoQ==
.
.
.
#EXT-X-CUE-IN
```

Enabling ad avail blanking in the output

You can enable ad avail blanking to blank out the content for an SCTE-35 message that is considered an ad avail (as defined by the ad avail mode in Getting ready: Set the ad avail mode).

A similar feature is blackout.

Blanking involves the following processing:

- Replace the video content associated with this event with an image that you specify or is with a black image.
- Remove the audio that is associated with this event.
- Remove the captions that are associated with this event.

Comparison to Manifest Decoration and Passthrough

Ad avail blanking applies to all outputs. You cannot choose to blank out for some outputs (for example, the HLS output) and not blank out for others (for example, the Microsoft Smooth output). It is an all-or-nothing decision.

Manifest decoration and passthrough have a smaller scope: they apply only to outputs that support these features.

Important

Be careful not to get into the following situation:

- You do *not* do passthrough.
- You do *not* do manifest decoration in a specific output (because they are not supported or because you choose not to).
- You do implement blanking

In this situation, there will be no markers for where the blanked content occurs. The only way to identify where this blanking is occurring will be to look for the IDR i-frames that identify where the SCTE-35 message used to be.

Topics

Enabling blanking

Follow this procedure if you want to enable the ad avail blanking feature.

To enable blanking

1. In the channel that you are creating, in the navigation pane, choose **General settings**.
2. Set the ad avail mode, if you have not already done so. See the section called "Get ready: Set the SCTE-35 source". The mode identifies which of all possible events are treated as triggers for blanking, which determines when video is blanked.
3. Still in **General settings**, in **Avail blanking**, in **State**, choose **Enabled**.
4. In **Avail blanking image**, choose the appropriate value:

   - Disable: To use a plain black image for blanking.
   - Avail blanking image: To use a special image for blanking. In the **URL** field, type the path to a file in an S3 bucket. For integration with MediaLive, the bucket name mustn't use dot notation. For example, `mycompany-videos` is acceptable but `mycompany.videos` isn't.

The file must be of type .bmp or .png. Also enter the user name and Systems Manager password parameter for accessing the S3 bucket. See the section called “About the feature for creating password parameters”.

Triggers for ad avail blanking

For ad avail blanking, the ad avail mode that you set controls which SCTE-35 events result in the blanking of the content.

Triggers in splice insert mode

This section describes which message type and segmentation type combination is blanked by ad avail blanking when the Ad Avail mode is Splice Insert mode.

Message type ID	Segmentation type	Does splice insert mode treat this message as an ad avail?
splice insert	No segmentation descriptor present	No
	Provider advertisement (0x30/0x31)	Yes, it treats it as an ad avail
	Distributor advertisement (0x32/0x33)	Yes, it treats it as an ad avail
	Provider placement opportuni ty (0x34/0x35)	Yes, it treats it as an ad avail
	Distributor placement opportunity (0x36/0x37)	Yes, it treats it as an ad avail
	Break (0x22/0x23)	Yes, it treats it as an ad avail
	Other: Programs, Chapters, Network, Unscheduled	No
time signal	No segmentation descriptor present	Not applicable to time signal messages

Message type ID	Segmentation type	Does splice insert mode treat this message as an ad avail?
	Provider advertisement (0x30/0x31)	Yes, it treats it as an ad avail
	Distributor advertisement (0x32/0x33)	Yes, it treats it as an ad avail
	Provider placement opportuni ty (0x34/0x35)	Yes, it treats it as an ad avail
	Distributor placement opportunity (0x36/0x37)	Yes, it treats it as an ad avail
	Break (0x22/0x23)	Yes, it treats it as an ad avail
	Other: Programs, Chapters, Network, Unscheduled	No

Triggers in timesignal APOS mode

This section describes which message type/segmentation type combination is blanked by ad avail blanking when the Ad Avail mode is Timesignal with APOS mode.

Message type ID	Segmentation type	Does timesignal APOS mode treat this message as an ad avail?
splice insert	No segmentation descriptor present	No
	Provider advertisement	No
	Distributor advertisement	No
	Placement opportunity	No
	Break	No

Message type ID	Segmentation type	Does timesignal APOS mode treat this message as an ad avail?
	Other: Programs, Chapters, Network, Unscheduled	No
time signal	No segmentation descriptor present	Not applicable to time signal messages
	Provider advertisement	No
	Distributor advertisement	No
	Placement opportunity	Yes, it treats it as an ad avail
	Break	Yes, it treats it as an ad avail
	Other: Programs, Chapters, Network, Unscheduled	No

Ad avail blanking restriction flags

Restrictions in the input

SCTE-35 messages of type time_signal always contain segmentation descriptors.

SCTE-35 messages of type splice_insert might or might not include segmentation descriptors.

If the input has SCTE-35 messages that do include segmentation descriptors, these segmentation descriptors always include two types of flags. Each flag has a value of "true" or "false" and provides additional information as guidance for blanking in specific situations:

- web_delivery_allowed_flag
  - True means that there is no restriction on including the ad avail event's content in a stream that is intended for web delivery: there is no need to blank out content in streams intended for web delivery.
  - False means there is a restriction: the content should be blanked out.
- no_regional_blackout_flag

(The wording of this flag is confusing. Think of it as the "regional_delivery_allowed_flag".)

- True means that there is no restriction on including the ad avail event's video in a stream that is intended for regional markets: there is no need to blank out content in streams intended for regional markets.
- False means there is a restriction: the content should be blanked out.

If neither flag is present (usually the case with splice_inserts), then both are considered to be false. Blanking should occur.

If both flags are present (which is usually the case with time_signal; it is unusual to have only one flag present), then a "false" for one flag takes precedence over a "true" for the other flag. Blanking should occur.

Typically, in any message in the input only one of these flags is ever set to false, so only one restriction is ever in place. There would typically never be *both* a regional delivery restriction and a web delivery restriction. This is because if content is considered restricted for regional delivery, then it would not also be considered restricted for web delivery (where the concept of a region makes no sense).

Representation of these Restrictions in MediaLive

There are two fields in MediaLive that let you control how MediaLive responds to the these flags. See the section called "Enabling blanking". Typically, you set the two fields to Follow (the default), to instruct MediaLive to follow the behavior implied by the value of the flag.

Enabling blackout in the output

You can enable blackout to blank out the content for an SCTE-35 message that is of type *other event* (as defined by the mode in Getting ready: Set the ad avail mode). For example, chapters and programs.

(A similar feature is described in the section called "Ad avail blanking".)

Blackout involves the following processing:

- Replace the video content associated with the event with an image that you specify or is with a black image.
- Remove the audio that is associated with the event.

- Remove the captions that are associated with the event.

Comparison to manifest decoration and passthrough

Blackout applies to all outputs. You cannot choose to black out for some outputs (for example, the HLS output) and not black out for others (for example, the Microsoft Smooth output). It is an all-or-nothing decision.

Manifest decoration and passthrough have a smaller scope: they apply only to outputs that support these features.

> **Important**
>
> Take note of this fact, because if you do *not* do passthrough and do *not* do manifest decoration in a specific output (because they are not supported or because you choose not to) but you do implement blanking, there will be no "markers" for where the blanked content occurs. The only way of identifying where this blanking is occurring will be to look for the IDR i-frames that identify where the SCTE-35 message used to be.

Topics

Enabling blackout

Follow this procedure if you want to enable the blackout feature.

To enable blackout

1. In the channel that you are creating, in the navigation pane, choose **General settings**.
2. Set the ad avail mode, if you have not already done so. See the section called “Get ready: Set the SCTE-35 source”. The mode identifies which of all possible events are treated as triggers for blackouts, which determines when video is blacked out.
3. Still in **General settings**, in **Blackout slate**, in **State**, choose **Enabled**.

4. For **Blackout slate image**, choose the appropriate value:

   - **Disable**: To use a plain black image for blackout.
   - **Avail blanking image**: To use a special image for blackout. In the **URL** field, enter the path to a file in an Amazon S3 bucket. For integration with MediaLive, the bucket name mustn't use dot notation. For example, `mycompany-videos` is acceptable but `mycompany.videos` isn't. The file must be of type .bmp or .png. Also enter the user name and Systems Manager password parameter for accessing the S3 bucket. For information about this key, see the section called “About the feature for creating password parameters”.
5. If you want to enable network end blackout (in other words, black out content when network transmission has ended and remove blackout only when network transmission resumes), continue reading. If you don't want to enable it, you have now finished setting up.
6. For **Network end blackout**, choose **Enabled**.
7. For **Network end blackout image**, choose the appropriate value:

   - **Disable**: To use a plain black image for blackout.
   - **Network end blackout image**: To use a special image for network end blackout. In the **URL** field, enter the path to a file in an Amazon S3 bucket. For integration with MediaLive, the bucket name mustn't use dot notation. For example, `mycompany-videos` is acceptable but `mycompany.videos` isn't. The file must be of type .bmp or .png. Also enter the user name and Systems Manager password for accessing the S3 bucket. See the section called “About the feature for creating password parameters”.
8. For **Additional settings**, in **Network ID**, type the EIDR ID of the network in the format 10.nnnn/xxxx- xxxx- xxxx- xxxx-xxxx-c (case insensitive). Only network end events with this ID will trigger blackout.

Triggers for blackout

The blackout feature is triggered only by time_signal messages of segmentation type **Other**. It is not triggered by splice_insert messages of any segmentation type, and is not triggered by time_signal messages of any type except **Other**.

SCTE-35 messages of type ID "splice insert" and messages of type ID "time signal" can both include "Other" time_signal messages. Therefore, when enabling blackout, the ad avail mode is not relevant. Blackout works the same with either mode.

The segmentation ID triggers blackout based on events, as shown in the following table.

Message type Id	SCTE-35 segmentation type	Start blacking out	End blacking out
splice insert	Any	Not a trigger	Not a trigger
time signal	Provider advertisement, Distributor advertisement, Break	Not a trigger	Not a trigger
	Chapter Start	Start blacking out	
	Chapter End		End blacking out
	Network *End*	Start blacking out	
	Network *Start*		End blacking out
	Program Start	Start blacking out	
	Program End		End blacking out
	Unscheduled Event Start	Start blacking out	
	Unscheduled Event End		End blacking out

For example, if the blackout feature is enabled, then blanking always occurs when a Program Start message is encountered and always ends when a Program End message is encountered.

Note that the triggers for blackout on a Network event are different from the other events:

- With Network, blanking starts when the Network *End* instruction is encountered.
- With other events, blanking starts when the "Event *Start*" instruction is encountered.

End event trigger hierarchy

Events have the *strength hierarchy* shown in the following table. A blackout can be ended only by an event of equal or greater strength than the event that started it.

For example, if the blackout is started by a Program Start, it can be ended by a Network Start, an Unscheduled Event End or a Program End. It cannot be ended by a Chapter End. MediaLive ignores the "end blackout" instruction implied by the Chapter End.

SCTE-35 segmentation type	Strength
Network	1 (Strongest)
Unscheduled Event	2
Program	3
Chapter	4 (Weakest)

Blackout restriction flags

Restrictions in the input

The segmentation descriptors in messages that are blackout triggers always include two types of flags. These flags provide additional information as guidance for blackout in specific situations:

- web_delivery_allowed_flag
  - True means that there is no restriction on including the event's content in a stream that is intended for web delivery. There is no need to black out content in streams intended for web delivery.
  - False means that there is a restriction. The content should be blacked out.
- no_regional_blackout_flag
  - True means that there is no restriction on including the event's video in a stream intended for regional markets. There is no need to black out content in streams intended for regional markets.
  - False means that there is a restriction. The content should be blacked out.

If both flags are present (which is usually the case; it is unusual to have only one flag present), then a "false" for one flag takes precedence over a "true" for the other flag. Blackout should occur.

Typically, in any message in the input only one of these flags is ever set to false, so only one restriction is ever in place. There would typically never be both a regional delivery restriction and

a web delivery restriction. This is because if content is considered restricted for regional delivery, then it would not also be considered restricted for web delivery (where the concept of a region makes no sense).

Representation of these flags in MediaLive

There are two fields in MediaLive that let you control how MediaLive responds to these flags. See the section called “Enabling blanking”. Typically, you set the two fields to Follow (the default), to instruct MediaLive to follow the behavior implied by the value of the flag.

Enabling SCTE-35 passthrough or removal

You can set up the MediaLive channel so that SCTE-35 messages from the input are passed through (included) in the data stream for the following outputs:

- Outputs in an Archive output group.
- Outputs in an HLS output group.
- Outputs in a MediaPackage output group. For these types of output groups, passthrough is always enabled. You can't disable it.
- Outputs in a UDP output group.
- Outputs in a Multiplex output group. For Multiplex output groups, SCTE-35 passthrough is enabled by default.

Alignment with video

The PTS of the SCTE-35 message is adjusted to match the PTS of the corresponding video frame.

Passthrough is at the output level

SCTE-35 passthrough or removal applies at the output level. The messages are passed through or removed only in a specific output. For most outputs, the default behavior (if you do not change the configuration fields) is to remove the messages. For MediaPackage outputs, the default behavior is to pass through the messages; you can't change this behavior.

Packet identifier (PID) selection from the input

If your source contains multiple SCTE-35 PIDs, you might want to select a specific PID to pass through to the output. By default, MediaLive will select the first SCTE-35 PID that is present on

the input. This can be changed by selecting a specific PID value from the **General input settings** section of the **Input attachment**. If the PID value selected is not present in the input, no SCTE-35 PID will be passed from input and an alert will be triggered.

Enabling passthrough for Archive outputs

Follow this procedure if you want to enable or disable passthrough for Archive outputs.

To enable passthrough

1. In the channel that you are creating, find the **Archive** output group that contains the output that you want to set up.
2. Choose that output.
3. In **PID settings**, complete the following fields:
   - **SCTE-35 control**: Set to **Passthrough**.
   - **SCTE-35 PID**: Leave the default PID or enter the PID where you want the SCTE-35 messages to go.
4. If appropriate, repeat for other outputs in this or other **Archive** output groups.

All SCTE-35 messages from the input are included in the data stream of the outputs that you have set up.

Enabling passthrough for CMAF Ingest outputs

Follow this procedure if you want to enable or disable passthrough for CMAF Ingest outputs.

To enable passthrough

1. In the **Create channel** or **Edit channel** page for the channel, in the **Channel** panel, find the **CMAF Ingest** output group that you want to set up. Select the output group by its name. The details appear in the right panel.
2. In **CMAF Ingest settings** section, set **SCTE35 type** to the appropriate value:
   - **NONE**: Omits the SCTE 35 messages from the output group.
   - **SCTE_35_WITHOUT_SEGMENTATION**: Includes (passes through) the SCTE 35 messages in the output group.

Each SCTE 35 message inserted will result in a new IDR in the video, but it won't result in a new segment. Note that CMAF Ingest doesn't require that SCTE 35 messages force a new segment.

Enabling passthrough for HLS outputs

Follow this procedure if you want to enable or disable passthrough for HLS outputs.

To enable passthrough

1. In the channel that you are creating, find the HLS output group that contains the output that you want to set up.
2. Choose that output.
3. In **PID settings**, complete the following fields:
   - **SCTE-35 behavior**: Set to **Passthrough**.
   - **SCTE-35 PID**: Leave the default PID or enter the PID where you want the SCTE-35 messages to go.
4. If appropriate, repeat for other outputs in this or other HLS output groups.

All SCTE-35 messages from the input will be included in the data stream of the outputs that you have set up.

Enabling passthrough for UDP outputs

Follow this procedure if you want to enable or disable passthrough for UDP outputs.

To enable passthrough

1. In the channel that you are creating, find the UDP output group that contains the output that you want to set up.
2. Choose that output.
3. In **PID settings**, complete the following fields:
   - **SCTE-35 control**: Set to **Passthrough**.
   - **SCTE-35 PID**: Leave the default PID or enter the PID where you want the SCTE-35 messages to go.

4. If appropriate, repeat for other outputs in this or other UDP output groups.

All SCTE-35 messages from the input will be included in the data stream of the outputs that you have set up.

Selecting SCTE-35 packet identifier (PID) from input

Follow this procedure if you want to select a specific SCTE-35 from the input.

To select the PID

1. In the channel that you are creating or editing, navigate to the **Input attachment** you would like to select a PID from.
2. Choose that **Input attachment**.
3. In **General input settings**, complete the following field:
   - **SCTE-35 PID**: Enter the PID value. If the value is left blank, the first SCTE-35 PID present in the input will be selected.
4. If appropriate, repeat for other **Input attachments**.

Inserting SCTE-35 messages using the schedule

Use the channel schedule to insert SCTE-35 messages in the content. For example, you can add an action in the channel schedule to insert a splice insert in the running channel at a specific time.

The main use case for this feature is to add SCTE-35 messages, when the source content doesn't already include SCTE-35 messages.

To insert SCTE-35 messages in the content, create actions in the schedule. For detailed information, see *Setup: Creating a schedule*.

After MediaLive inserts the SCTE-35 message in the channel, MediaLive processes the message in the same way as it would process SCTE-35 messages that were in the input. You define this processing when you create the channel and configure these options:

- Blanking
- Blackout
- Manifest decoration

- Passthrough

For a summary of these options, see the section called "Scope of processing by feature" and the section called "Supported features by output type".

POIS signal conditioning

You can configure an AWS Elemental MediaLive channel so that your POIS server can perform *signal conditioning* on SCTE-35 messages that are in the content. Each time MediaLive encounters a SCTE-35 message in the content, MediaLive sends the message to the POIS server. The POIS server sends back a response to create a new SCTE 35 message, to replace the original message with different content, to delete the existing message, or to do nothing.

> **Note**
>
> To implement POIS signal conditioning, your organization must have access to a POIS server.

Topics

Supported version of the specification

MediaLive communicates with a POIS server using the ESAM API. MediaLive follows the following version of the ESAM specification:

OpenCable Specifications Alternate Content Real-time Event Signaling and Management API, OC-SP-ESAM-API-I03-131025

About POIS signal conditioning

You can configure a MediaLive channel so that your POIS server handles SCTE-35 message that are in the content.

> **Note**
>
> To implement POIS signal conditioning, your organization must have access to a POIS server.

Supported conditioning actions

Each time MediaLive encounters a SCTE-35 message in the content, MediaLive sends the message to the POIS server. The POIS server responds in one of these way:

- Replace: It replaces the content of the original SCTE-35 message and sends it to MediaLive. The output will contain only the original SCTE 35 message, but with the new content.
- Delete: It instructs MediaLive to delete the SCTE 35 message. The output won't contain the original SCTE 35 message.
- No op: It instructs MediaLive to do nothing. The output will contain the original SCTE 35 message, with the original content.

The SCTE 35 messages that the POIS server returns are completely compliant with the SCTE 35 standard.

Number of channels and number of POIS servers

- Each MediaLive channel can communicate with only one POIS server.
- One POIS server can communicate with multiple MediaLive channels. In this case, the POIS server identifies each channel using a unique combination of a POIS acquisition point identity and a zone identity.

POIS signal conditioning and standard channels

If the channel is a standard channel (with two pipelines), then each pipeline sends the SCTE 35 message to the POIS server. The POIS server responds to each request. Each pipeline handles its own response. Typically, the POIS server sends the identical instruction to both pipelines.

Setting up for POIS signal conditioning

With POIS signal conditioning, the MediaLive channel and the POIS server must be set up with identical information.

Required information

Obtain the following information from the POIS operator:

- The *POIS server endpoint*. This is the URL of the POIS server that MediaLive will send events to. The URL must be reachable by MediaLive.
- The *Acquisition point identity* and the *Zone identity* (optional). These two fields ensure that MediaLive and the POIS server have a common identifier for the channel.
- POIS endpoint credentials, if the POIS server requires credentials.

Set up the channel

You must configure the channel with information about the POIS server.

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. Display the details for the channel. In the **General settings** section, expand **Avail Configuration**.
3. Complete the fields as follows:
   - **Avail Settings**: **ESAM**
   - **POIS Endpoint**: The URL that you obtained from the POIS operator.
   - **Acquisition Point Identity**: The value that you obtained from the POIS operator.
   - **Zone Identity**: The value that you obtained from the POIS operator.
   - **Ad Avail Offset**: Enter 0 unless the POIS operator tells you to enter a different value.
   - **POIS Endpoint Credentials** (optional): Complete these fields if your POIS server requires a username and password.

Sharing encodes among outputs

You can share a single encode among several outputs within one channel. You can clone an encode so that it serves as the basis for a new encode within the channel.

Sharing encodes

You can share an encode among several outputs when you want these outputs to have identical encodes. When you share an encode, there is only one instance of the encode in the channel. All the affected outputs use that encode.

For example, you might have a channel containing these output groups:

- One Archive output group, with one video encode.
- One HLS output group, with three video encodes in an ABR stack.
- One RTMP output group, with one video encode.

The video encode in the RTMP output group might be identical to one of the video encodes in the HLS output group. Therefore, instead of creating five video encodes, you would create four encodes. You would set up the RTMP output group to share the appropriate encode from the HLS output group.

Encode sharing applies to video, audio, and captions. Sharing encodes reduces the effort of filling in fields. It also reduces the risk of error when you intend to create identical encodes among outputs. There is no chance that you will accidentally complete one field differently.

These rules apply to encode sharing:

- When you share an encode, you share all the fields, including the source selector for the encode.

  If you want to create two encodes that share all their configuration fields but are based on different sources, you *can't* share the single encode instance. You should instead clone the encode.
- You can share any encode among as many outputs as you want.
- You can share more than one encode in the channel.
- You can share only within the same channel. You can't share across channels.

For instructions to clone encodes when you are creating a channel, see the section called “Set up video”, the section called “Set up audio”, and the section called “Set up captions”.

Cloning encodes

You can clone an encode so that it serves as the basis for a new encode within the channel.

For example, you might have two audio encodes in the channel that share some fields. You create the first audio encode. You then create a second encode by cloning the first encode, and change any of the fields in the second encode. In this case, the two encodes are separate instances.

These rules apply to encode sharing:

- When you clone an encode, there are two instances of the encode, even if you don't change any fields in the second encode.
- After you have cloned the encode to create the new instance, you can change any of the fields, including the source selector for the encode.
- You can combine sharing and cloning. For example, you could clone encode A to create encode B. You could then share encode B among two or more outputs.
- You can clone more than one encode in the channel.
- You can clone only within the same channel. You can't clone across channels.

For instructions to clone encodes when you are creating a channel, see the section called “Set up video”, the section called “Set up audio”, and the section called “Set up captions”.

Handling SMPTE 2038 metadata

You can configure MediaLive to extract specific ancillary data from a SMPTE 2038 stream contained in these inputs:

- Elemental Link inputs
- HLS transport stream (TS) inputs
- MediaConnect inputs
- RTP inputs

> **Note**
>
> This section assumes that you are familiar with creating or editing a channel, as described in *Setup: Creating a channel* .

Topics

- How MediaLive uses the SMPTE 2038 stream
- Configuring outputs for KLV metadata

Metadata that MediaLive can extract

MediaLive can extract the following data from a SMPTE 2038 stream that is in the source.

Captions

- ARIB captions – Captions that are compliant with ARIB STD-B37 version 2.4.
- Embedded captions – Captions carried as ancillary captions that are compliant with SMPTE 334. The ancillary captions themselves must be compliant with EIA-608 standard (also known as CEA-608 or "line 21 captions") or CEA-708 standard (also known as EIA-708).
- Teletext captions – OP47 teletext format, also known as SMPTE RDD-08 (compliant with ITU-R BT.1120-7).

Timecode

- Timecode – A SMPTE 12M timecode. MediaLive recognizes this timecode as an embedded timecode source.

Ad avail messages

- SCTE 104 messages.

Metadata

- KLV metadata – Data that is compliant with SMPTE 336M-2007.

A well-formed SMPTE 2038 stream

For MediaLive to extract and process the data appropriately, the SMPTE 2038 stream in the input must meet certain criteria:

- The SMPTE 2038 stream must be present in every PMT.

- The PID in which the SMPTE 2038 stream is located must not change in the stream. There is no support for changing the PID and sending a new PMT identifying that PID.
- The transport stream should contain the SMPTE 2038 stream in only one PID. If it's present in more than one PID, there's no guarantee that MediaLive will identify the PID that appears first. It could choose another PID, with results you don't intend.

Note that if the input is a Elemental Link input, embedded captions (if any), the timecode, and KLV metadata (if any) are always in a SMPTE 2038 stream. The stream is always well-formed.

Configuring inputs

If you want MediaLive to use the data in a SMPTE 2038 stream, you must configure the input to read the SMPTE 2038.

1. On the **Create channel** page, find the **Input attachment** for the relevant input.
2. In **General input settings**, set **Prefer SMPTE 2038** to one of the following:
   - **Prefer** – For a specific item of data, MediaLive first looks for the data in a SMPTE 2038 PID. If the data is not found in the SMPTE 2038 stream or if there is no SMPTE 2038 stream, MediaLive looks for the data in other locations in the stream.
   - **Ignore** (default) – MediaLive never looks for a SMPTE 2038 stream. Even if a specific item of data is not available in other places in the stream, MediaLive doesn't look for a SMPTE 2038 stream. For example, you might set the timecode source to Embedded (in the **General Configuration** section for the channel). With **Ignore**, if the timecode source isn't in the video stream, MediaLive won't look for it in a SMPTE 2038 stream.

   Note that with Elemental Link input, any KLV metadata is always in a SMPTE 2038, never in a different PID. Therefore, if you have been told that the source includes KLV metadata, always choose **Prefer**.

How MediaLive uses the SMPTE 2038 stream

If you set up to prefer SMPTE 2038 in an input, MediaLive uses the data according to the following rules.

Captions

You might set up the input with captions selectors that specify **ARIB**, **Embedded**, or **Teletext**. In this case, MediaLive first looks for the specified type of captions in the SMPTE 2038 stream. If MediaLive doesn't find the captions there, it looks in other locations in the stream.

Regardless of where MediaLive finds the captions, MediaLive extracts them, and processes them in the usual way, according to how you set up for captions in the output.

Timecode

When an input includes a SMPTE 2038 stream, MediaLive first looks for a SMPTE 12M timecode in the SMPTE 2038 stream. If MediaLive doesn't find the timecode there, it looks for a timecode embedded directly in the video stream. MediaLive associates the SMPTE 12M timecode with the closest video frame.

For information about how MediaLive uses the timecode, see the section called “How the output timecode works at runtime ”.

Ad avail messages

If you prefer SMPTE 2038 in an input, MediaLive extracts any SCTE 104 messages it finds, then immediately converts them to SCTE 35 messages. You can then handle the messages as you would handle SCTE 35 messages from any source. For more information, see the section called “SCTE-35 message processing”.

KLV metadata

If you prefer SMPTE 2038 in an input, MediaLive extracts any KLV data that it finds.

You can choose to pass through the KLV metadata in one or more of the following output groups. MediaLive wraps the KLV in a SMPTE 2038 stream.

- Archive
- MediaPackage
- HLS (with a TS container)
- UDP/TS

The setup steps follow.

Configuring outputs for KLV metadata

You can choose to pass through the KLV metadata in one or more of the following output groups.

> **Note**
>
> The information in this section assumes that you are familiar with the general steps for creating a channel, as described in *Setup: Creating a channel* .

Archive

1. On the **Create channel** page, in the **Output groups** section, in the **Archive** group, choose the output.
2. In **Output settings**, select **Container settings**, then select **PID settings**.
3. Set these fields:
   - **KLV**: Choose **PASSTHROUGH**
   - **KLV data PIDs**: Enter the PID where you want the KLV metadata.

MediaPackage

MediaPackage outputs are automatically set up for passthrough. If MediaLive finds KLV metadata in an input, it passes it through in a MediaPackage output, in PID 501.

HLS

You can pass through KLV metadata in any output that has a standard HLS container (a TS container).

1. On the **Create channel** page, in the **Output groups** section, in the **HLS** group, choose the output.
2. In **Output settings**, make sure that **HLS settings** specifies **Standard HLS**.
3. In **HLS settings**, select **PID settings**.
4. Set these fields:
   - **KLV**: Choose **PASSTHROUGH**
   - **KLV data PIDs**: Enter the PID where you want the KLV metadata.

UDP/TS

1. On the **Create channel** page, in the **Output groups** section, in the **UDP** group, choose the output.
2. In **Output settings**, select **Network settings**, then select **PID Settings**.
3. Set these fields:
   - **KLV**: Choose **PASSTHROUGH**
   - **KLV data PID**: Enter the PID where you want the KLV metadata.

Using ACLs for delivery to Amazon Simple Storage Service

In a channel, you might have one or more outputs where the destination is a bucket in Amazon Simple Storage Service (Amazon S3). If the bucket is owned by another AWS account (another organization), you typically want the other account to become the owner of the output files.

You can transfer ownership by setting up MediaLive to include a specific access control list (ACL) when delivering to the bucket.

For more information on preparing to use an ACL, see the following:

- For an Archive or Frame capture output – the section called “Controlling access to the output”
- For an HLS output – the section called “Controlling access to the output”

On the console, the field for enabling the feature is in the section for each output group. The field is described in the following sections:

- For Archive outputs – the section called “Destination fields”
- For frame capture outputs – the section called “Destination fields”
- For HLS outputs – the section called “Destination fields – Amazon S3”

Tagging AWS Elemental MediaLive resources

A *tag* is a metadata label that you assign or that AWS assigns to an AWS resource. Each tag consists of a *key* and a *value*. For tags that you assign, you define the key and value. For example, you might define the key as `stage` and the value for one resource as `test`.

Tags help you do the following:

- Identify and organize your AWS resources. Many AWS services support tagging, so you can assign the same tag to resources from different services to indicate that the resources are related. For example, you could assign the same tag to an AWS Elemental MediaLive channel and an endpoint that you assign to an AWS Elemental MediaTailor configuration.
- Track your AWS costs. You activate these tags on the AWS Billing and Cost Management dashboard. AWS uses the tags to categorize your costs and deliver a monthly cost allocation report to you. For more information, see Use Cost Allocation Tags in the AWS Billing User Guide.

The following sections provide more information about tags for AWS Elemental MediaLive.

Supported resources in AWS Elemental MediaLive

The following resources in AWS Elemental MediaLive support tagging:

- Channels
- Inputs
- Input security groups
- AWS Elemental Link devices
- Multiplexes
- Reservations

For information about adding and managing tags, see Managing tags.

Tag restrictions

The following basic restrictions apply to tags on AWS Elemental MediaLive resources:

- Maximum number of tags that you can assign to a resource – 50
- Maximum key length – 128 Unicode characters
- Maximum value length – 256 Unicode characters
- Valid characters for key and value – a-z, A-Z, 0-9, space, and the following characters: _ . : / = + - and @
- Keys and values are case sensitive
- Don't use `aws:` as a prefix for keys; it's reserved for AWS use

Additionally, AWS Elemental MediaLive doesn't support the tag-based access control feature of AWS Identity and Access Management (IAM).

Managing tags

Tags are made up of the `Key` and `Value` properties on a resource.

You can use the AWS Management Console to manage tags. You can also use the AWS Elemental MediaLive console, the AWS CLI, or the MediaLive API to add, edit, or delete the values for these properties.

Tagging using the AWS Management Console

We recommend that you manage tags by using the Tag Editor on the AWS Management Console. The Tag Editor provides a central, unified way to create and manage your tags. The Tag Editor provides the best results, including consistency between tags within MediaLive and between MediaLive and other services.

For more information, see Working with Tag Editor in Getting Started with the AWS Management Console.

Tagging using MediaLive

For information about managing tags using the MediaLive console, see the following:

- the section called “Channel and input details” – for information about including tags when you create a channel
- the section called “Editing and deleting a channel” – for information about modifying tags in an existing channel
- the section called “Inputs” – for information about including tags in an input
- the section called “Input security groups” – for information about including tags in an input security group
- the section called “Creating a multiplex and program” – for information about including tags in a multiplex

For information about managing tags using the MediaLive API, see the following:

- Resources in the *AWS Elemental MediaLive API Reference*

Viewing input thumbnails in AWS Elemental MediaLive

AWS Elemental MediaLive can generate thumbnails for the video from inputs in your channels. You can view the thumbnails for each channel on the MediaLive console. You can also use one of the AWS APIs to work with thumbnails programmatically.

How thumbnails are generated

When you have enabled thumbnails in a channel and the channel is running, MediaLive generates a JPEG thumbnail every 2 seconds. The thumbnail exists for only 2 seconds, until it gets replaced by the next thumbnail. Each input has its own thumbnail, which means that MediaLive generates one thumbnail for a single-pipeline channel, and two thumbnails for a standard channel.

As soon as the thumbnail is generated, MediaLive displays it on the console, in the channel details page. It also makes the thumbnail available as binary data. You can use an AWS API to work with the binary data programmatically.

Encryption of the thumbnail

MediaLive always encrypts each thumbnail as it is created.

Topics

Enabling thumbnails in a channel

You must enable the thumbnails feature in each channel.

You can enable or disable thumbnails only when the channel is idle (not running).

Topics

- Enabling thumbnails on the console
- Enabling thumbnails programmatically

Providing IAM access

For the thumbnails feature to work, MediaLive needs access to Amazon S3:

- If your organization uses the MediaLiveAccessRole trusted entity, go to the **Channel and input details** page of the channel configuration, and look in the **General info** section. If the **Update role** button appears in this section, select the button. If the button doesn't appear, then the trusted entity already has the access that it needs
- If your organization uses custom trusted entity roles, then an IAM administrator must update the appropriate trusted entity roles. For information about the operations to add, read the section called "Reference: summary of trusted entity access". Search for **`thumbnails`** on that page. For information about how to update the role, see the section called "Step 2: Set up with complex option".

Enabling thumbnails on the console

> **Note**
>
> This section assumes that you are familiar with creating or editing a channel, as described in *Setup: Creating a channel* .

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. On the **Create channel** page or **Edit channel** page, choose **General settings**, then open the **Thumbnail configuration** section.
3. Select **Enable thumbnail configuration**. In **State**, choose **AUTO** or **DISABLED**.

Enabling thumbnails programmatically

To enable the thumbnails feature, include the `ThumbnailConfiguration` group of parameters in the JSON for the channel. Set the `State` parameter to `AUTO` (to enable) or `DISABLED`.

The following example shows the relative location of the parameters in the JSON for the channel.

```
{
  "ChannelClass": "SINGLE_PIPELINE",
  .
  .
  .
  "EncoderSettings": {
    .
    .
    .
    "TimecodeConfig": {
      "Source": "EMBEDDED"
    },
    "ThumbnailConfiguration": {
      "State": "DISABLED"
     },
    .
    .
    .
```

Viewing thumbnails on the console

MediaLive generates thumbnails for the currently active input in a channel that is running. For a standard channel, MediaLive generates two thumbnails. For a single-pipeline channel, MediaLive generates one thumbnail.

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. Choose **Channels** from the navigation bar. On the list of channels, select the channel by its name. The Details page appears.

   The **Status** section includes a thumbnail frame. If the channel has two inputs, the screen includes a tab for each input.

When the channel is running and the Details page is displayed, the thumbnail automatically updates every 2 seconds, for the pipeline in the active tab. MediaLive doesn't generate any thumbnails if this page isn't displayed.

If the channel isn't running, the frame is black.

If the channel stops running, the thumbnail preview stops updating. After a few seconds, the current thumbnail is replaced by a message.

Retrieving thumbnails programmatically

MediaLive generates thumbnails for the currently active input in a channel that is running. For a standard channel, MediaLive generates two thumbnails. For a single-pipeline channel, MediaLive generates one thumbnail.

You can use the AWS CLI to work with thumbnails programmatically. The following information assumes that you are familiar with the basics of using the AWS CLI. For information about the basics, see the AWS CLI Command Reference.

Use the `DescribeThumbnails` command. This command is represented differently in different interfaces:

- In the AWS CLI, the command is `describe-thumbnails`.
- In the API, the command is represented by an `HTTP GET` on `describe-thumbnails/kmsKeyId`.
- In the AWS SDKs, the command is represented by constructs that are suitable to that SDK language.

To retrieve thumbnails using the AWS CLI

1. Make sure that you have enabled thumbnails in the channel, and make sure that the channel is running.
2. Enter this command:

   aws medialive describe-thumbnails --channel-id *value* --pipeline-id *value* --thumbnail-type *value*

   Where:

   `channel-id` is required.

   `pipeline-id` is 0 or 1. If you want thumbnails for both pipelines, enter the command twice.

   `thumbnail-type` is always `CURRENT_ACTIVE`. This option is required, even though it has only one value.
3. The response appears on the screen. For example:

```
{
```

```
    "ThumbnailDetails": [
        {
            "PipelineId": "0",
            "Thumbnails": [
                {
                    "Body"base64 string of the JPEG image",
                    "ContentType": "image/jpeg",
                    "ThumbnailType": "CURRENT_ACTIVE",
                    "TimeStamp": "2023-07-15T21:01:11"
                }
            ]
        }
    ]
}
```

If thumbnails are disabled, the response looks like this:

```
{
    "ThumbnailDetails": []
}
```

Limit on thumbnails

There is a limit to the number of thumbnails that you can view or retrieve. The limit is:

`A number of API transactions per second, per account, in one Region`

The transaction limit is shared by all thumbnails — those that you display on the console, and those that you retrieve using an AWS API. For the current limit, see the MediaLive page in the Service Quotas console.

On the console, a thumbnail is generated for a channel only when the channel details page is displayed, and only in the active tab (meaning only for one pipeline in the channel). For the relevant pipelines, MediaLive makes a call to the API approximately every 2 seconds.

Timecodes

MediaLive has timecodes for the input pipeline and the output pipeline. The two timecodes are separate from each other. You can't configure the input timecode. You can configure the behavior

of the output timecode. You can also configure output to include the output timecode as metadata and/or to burn the output timecode into the video frame.

Topics

About timecodes and timestamps

MediaLive has timecodes for the input pipeline and the output pipeline. The two timecodes are separate from each other.

Input timecode

MediaLive has features that work only if incoming frames include embedded timecodes. These features include pipeline locking and watermarking. If an input doesn't have an embedded timecode, MediaLive won't implement the feature. For example, with pipeline locking, the pipelines won't get locked in a frame accurate way. (For more information about how the timecode affects pipeline locking, see the section called “Pipeline locking (output locking)”.

The input timecode source is not configurable.

Output timecode

MediaLive implements SMPTE timecode, which means that MediaLive assigns a timecode of the format `HH:MM:SS:FF` to each outgoing frame. The timecode rolls over at midnight.

There are three ways to initialize the output timecode in a channel:

- Embedded (the default): Use the embedded timecode to initialize the output timecode. MediaLive uses the timecode in the first frame that it ingests in the input. If the input does not contain a timecode, MediaLive uses UTC.
- UTC: Initialize the output timecode to the UTC time at the moment that the first frame enters the output side of the pipeline.
- Zero-based: Initialize the output timecode to 00:00:00:00.

The output timecode is used in features such as the PDT for an HLS output, and for the timecode for ID3 metadata that you might choose to include. You can also configure output to include the output timecode as metadata and/or to burn the output timecode into the video frame.

You can also configure output video to include the output timecode as metadata, and/or to burn the output timecode into the video frame.

Timestamps

MediaLive attaches a timestamp to all output content. Downstream systems use the timestamp for synchronization. The timestamp is a value such as the number of 90 KHz clock cycles.

Don't conflate timestamps and timecodes. They are different.

Configuring the start time for the output timecode

You can configure the start time for the output timecode.

> **Note**
>
> This procedure assumes that you are familiar with creating or editing a channel, as described in *Setup: Creating a channel* .

1. On the **Create Channel** page, in the **General settings** section, choose **Timecode configuration**.
2. In **Source**, choose the option for initializing the timecode in the output:
   - **EMBEDDED** – Use the timecode embedded in the source video.

     MediaLive looks for the timecode in the source video, as follows:
     - H.264 – A timecode inserted in an SEI message of type pic_timing, in accordance with section D.1.2 of ISO/IEC 14496-10-2005
     - H.265 – A timecode inserted in an SEI message of type timecode, in accordance with section D.2.26 of ITU-T H.265
     - MPEG2 – A timecode inserted in each GOP header, in accordance with section 6.2.2.6 of ISO/IEC 13818-2-2000 (R2006)
   - **SYSTEMCLOCK**– Use the UTC time.
   - **ZEROBASED** – Use 00:00:00:00.

3. (Optional) In **Sync threshold**, enter a threshold (in frames) for synchronizing the output timecode to the input timecode. For information about this field, see the section called “About the synchronization threshold”.

How the output timecode works at runtime

Initial channel start or restart

When you start the channel, the channel establishes the start timecode for the output pipeline:

- The channel samples the input timecode, if you set up the start time to reference the embedded timecode. If MediaLive doesn't find an embedded timecode in the source, it falls back to UTC.
- Or it sets the timecode to the current UTC time.
- Or it sets the timecode to 00:00:00:00.

The channel generates a new timecode for each output frame that it produces.

Input switches

When the channel switches to a different input, MediaLive doesn't reinitialize the timecode. Therefore, the output timecode isn't disrupted by an input switch.

Pausing and unpausing

If you pause the channel, MediaLive continues to encode frames, which it immediately discards. But because MediaLive continues to encode, the timecodes continue to increment. Therefore, when you unpause, there will be a timecode discontinuity in the output.

About the synchronization threshold

The timecode **Sync threshold** field synchronizes the output timecode with the input timecode. Drift can occur in several ways. For example, processing issues can occur that cause MediaLive to drop or repeat frames to compensate. Or there might be discontinuities in the input timecode stream.

Purpose of synchronization

Synchronization is useful if it is important for your workflow that the output timecode (that MediaLive generates) match the original input timecode.

- Matching might be important if you know that the downstream system must identify specific frames.

  Typically, the downstream system has already identified these frames based on the original input timecode. Therefore, the output timecode must match the original input timecode, in order for the downstream system to find the desired frame.
- Matching isn't important if the main purpose of the output timecode is simply to uniquely identify each output frame.

How synchronization works

After the input timecode and the output timecode have drifted apart by the specified number of frames, MediaLive inserts a discontinuity in the output timecode sequence, and sets the output timecode to match the current input timecode.

The main drawbacks of synchronizing are that it introduces timecode discontinuities into the metadata, and that it can't guarantee that each output timecode is unique.

Including timecode metadata in the output

You can set up the channel to include timecode metadata in the individual output encode. Timecode metadata is supported in any type of output group except Frame Capture.

The timecode is inserted according to the standard for the output encode:

- H.264 – The timecode is inserted in an SEI message of type pic_timing, in accordance with section D.1.2 of ISO/IEC 14496-10-2005
- H.265 – The timecode is inserted in an SEI message of type timecode, in accordance with section D.2.26 of ITU-T H.265
- MPEG – The timecode is inserted in each GOP header, in accordance with section 6.2.2.6 of ISO/IEC 13818-2-2000 (R2006)

To include timecode metadata in the output

On the output side, in each video encode, you specify whether to include the timecode. By default, the timecode is not included in the video encode.

1. On the **Create Channel** page, in the **Output groups** section, choose an output group, then choose an output.

2. Display the **Stream settings** section, and then choose the **Video** section. In **Codec settings**, choose the codec for this video encode. More fields appear.
3. Choose **Timecode**, then in **Timecode insertion**, choose an option:

   - **DISABLED** – This encode won't include timecode metadata.
   - **PIC_TIMING_SEI** (for H.264 or H.265) or GOP_timecode (MPEG) – This encode will include timecode metadata.

Burning the timecode into output

You can set up an encode to burn in the output timecode. The time code will become part of the video.

Note that the timecode burnin feature is independent of the timecode metadata feature. You don't have to enable timecode metadata in order to burn in the timecode.

To burn the timecode into the video output

1. On the **Create Channel** page, in the **Output groups** section, choose an output group, then choose an output.
2. Display the **Stream settings** section, and then choose the **Video** section. In **Codec settings**, choose the codec for this video encode. More fields appear.
3. Choose **Timecode**, then in **Timecode burn-in settings**, choose **Timecode burnin**. More fields appear.
4. Set the style and position of the timecode in the video frame. In the optional **Prefix** field, enter any descriptor. For example, **`UTC-1`**.

Implementing a trick-play track

Trick-play is used in digital video players to mimic some capabilities of analog players, including fast-forward and rewind capabilities. These capabilities often include a trick-play *track*—a visual cue for the person using the video player. In AWS Elemental MediaLive, you can include track assets in the output group. The downstream system for that output group can use these assets to implement the visual cue in their trick-play implementation.

MediaLive provides two methods for including these assets:

- An I-frame-only manifest that conforms with the HLS specification.
- A trick-play track that conforms with the Image Media Playlist specification, version 0.4.

MediaLive supports these methods as follows:

- In HLS output groups, MediaLive supports both methods.
- In MediaPackage output groups, MediaLive supports trick-play via the Image Media playlist specification

Choosing an implementation of trick-play track

You can follow one or both trick-play methods in the same output group.

Before you follow either method, contact the downstream system for the output group to find out how they implement trick-play. Find out the following:

- Can the downstream system support a trick-play track? If so, which trick-play specification does it follow?
- Is the supported implementation required or optional? Both of these implementations introduce specific lines into the HLS manifest. If the lines are absent, the downstream system will fail to handle the output from MediaLive?

  It is likely that the downstream system considers both of these implementations to be optional.
- If you choose the I-frame-only manifest method, confirm that the downstream system supports the method according to the HLS specification. If the downstream system has a variation, it's possible that the downstream system won't be able to handle the output from MediaLive. MediaLive doesn't support customizations of the method.
- If you choose the image media playlist method, confirm that the downstream system supports the method according to the Image Media Playlist specification. If the downstream system has a variation, it's possible that the downstream system won't be able to handle the output from MediaLive. MediaLive doesn't support customizations of the implementation.

Topics

Trick-play track via I-frames

In an HLS output group, you can support trick-play track by providing an I-frame-only manifest.

How the method works

When you create the HLS output group, you create one or more video outputs, in the usual way. For a reminder of the output group structure, look at the diagrams in the section called "HLS or MediaPackage output group". In the output group, you enable the field to create an I-frame-only manifest that conforms to the HLS specification.

MediaLive produces two child manifests for each encode—one manifest for handling the video in the usual way, and the I-frame-only manifest. The I-frame-only manifest lets the downstream player identify specific video frames to request, to construct the trick-play track. So this trick-play track method doesn't produce additional encodes in the output group.

Each I-frame-only manifest contains the following:

- One `#EXT-X-I-FRAMES-ONLY` tag, to indicate that the manifest is I-frame-only.
- Many `#EXT-X-BYTERANGE` entries. Each entry identifies the position of an I-frame position.

Setting up

You set up the trick-play track once for the entire HLS output group.

> **Note**
>
> The information in this section assumes that you are familiar with the general steps for creating a channel.

To set up an I-frame-only manifest

Include these steps when you create the HLS output group.

1. In the **HLS output group**, in **Manifest and segments**, for **I-frame only playlists**, choose **ENABLED**.
2. Set up the remaining fields in the output group as you normally would. Set up the video, audio, and captions outputs and encodes as you normally would.

Trick-play track via the Image Media Playlist specification

In an HLS or MediaPackage output group, you can support a trick-play track by providing an asset that follows the Image Media Playlist specification, version 0.4. The MediaLive implementation follows the time-based method of the specification. The specification is located here:

https://github.com/image-media-playlist/spec/blob/master/image_media_playlist_v0_4.pdf

Roku is one example of a platform that implements this specification.

How the method works

When you create the output group, you create standard outputs in the usual way for the video, audio, and captions encodes. See the section called “HLS or MediaPackage output group” for diagrams that illustrate the structure of the encodes in the output group.

You also create one output that contains one frame capture encode. The encode is a series of JPEG files, one file for every video segment, which means that the capture follows the segmentation of the video encode. This encode is the asset that the downstream player can use to implement the trick-play track.

MediaLive creates a main manifest and child manifests in the usual way. The main manifest includes an `EXT-X-IMAGE-STREAM-INF` tag for the frame capture encode. The child manifest for the frame capture encode contains `EXT-X-IMAGES-ONLY` tags. The contents and format of these tags comply with the Image Media Playlist specification.

Setting up

You set up the trick-play track in the output group by creating an additional output that contains a video encode consisting of frame captures. You can add up to three frame capture outputs in one output group, and up to three frame capture encodes in the channel.

> **Note**
>
> The information in this section assumes that you are familiar with the general steps for creating a channel.

To set up the frame capture encode in an HLS output group

To create a frame capture encode in an HLS output group, you create a special type of output and set its video codec to **Frame Capture**.

1. In the **HLS output group**, in **HLS outputs**, choose **Add output** to add another output.
2. For that output, choose **Settings**, and in **Output settings**, set **HLS settings** to **Frame capture hls**.
3. In **Stream settings**, choose **Video** and set up the video fields, including:
   - **Width** and **Height** – Contact your downstream system to obtain the correct values. If you guess at the values, the experience on the downstream player might not be optimal.
   - **Codec settings** – Choose **Frame capture**.
   - **Capture interval** – Don't change the value of this field. Leave it empty, so that the frame capture uses the default interval.
4. Choose **Audio 1** and choose **Remove audio** so that the container has only one encode (a video encode).

To set up the frame capture encode in a MediaPackage output group

To create a frame capture encode in a MediaPackage output group, you create a regular output and set its video codec to **Frame Capture**.

1. In the **MediaPackage output group**, in **MediaPackage outputs**, choose **Add output** to add another output.
2. For that output, choose **Settings**, and then choose **Stream settings**. In **Stream settings**, choose **Video**.
3. In **Codec settings**, choose **Frame capture**.
4. Set up the other video fields, including:
   - **Width** and **Height** – Contact your downstream system to obtain the correct values. If you guess at the values, the experience on the downstream player might not be optimal.
   - **Capture interval** – Don't change the value of this field. Leave it empty, so that the frame capture uses the default interval.
5. Choose **Audio 1** and choose **Remove audio** so that the container has only one encode (a video encode).

The output is part of the ABR stack and has the same destination as the other encodes in the HLS or MediaPackage output group.

Handling a straightforward color space conversion

You can control how MediaLive takes the color space and color space metadata in a video source and manipulates it in the video output. You can set up each output video encode to convert or pass through the color space, and to include or omit the color space metadata.

All video belongs to a specific color space. The color space defines the range of color for the video. Video can include color space metadata that provides information about the color space. When metadata is missing, the video still has a color space, but it is impossible for MediaLive to manipulate the color space.

Default behavior

The default behavior is to pass through the color space and pass through the color space metadata.

Topics

Determine if this section applies to your channel

In this guide there are two sections about handling color space — this *straightforward handling* section, and the section called “Video – complex color space conversion”.

The current section provides procedures you can follow if the input color spaces and color space metadata are all *clean*. The procedures in this section are shorter than those in the other section.

To determine if your content meets the requirements for these procedures, read the following table. Each row in the table describes a different scenario that this *straightforward handling* section covers. Find the scenario that applies to your content. If none of these scenarios applies to you, then you must you must read the section called “Video – complex color space conversion”.

Type of handling in the channel	Characteristics of color space	Characteristics of metadata in input
You are passing through the color space in every output.	The color space can be any color space. It doesn't have to be a color space that MediaLive can to convert from or can convert to.	The color space metadata must either be correct, or you must be prepared to remove it from the output.
You are converting the color space in at least one output. You might be passing through the color space in other outputs.	If converting, the color space or color spaces must be one of the color spaces that MediaLive can convert. The color space can change within one source, but it must meet the requirement.	The color space metadata must be present and must match the color space.
	If passing through, the source color space can be any color space. It doesn't have to be a color space that MediaLive can convert from or can convert to.	The color space metadata must either be correct, or you must be prepared to remove it from the output.
You are converting the color space in at least one output, and you are using 3D LUT files.	If converting, the color space or color spaces must be one of the color spaces that MediaLive can convert. The color space can change within one source, but it must meet the requirement.	The color space metadata must be present and must match the color space. We assume that if you are using 3D LUT files, the content is well formed. Use of 3D LUT files is documente

Type of handling in the channel	Characteristics of color space	Characteristics of metadata in input
		d only in this section. (It isn't documented in the section called "Video – complex color space conversion".)
	If passing through, the color space can be any color space. It doesn't have to be a color space that MediaLive can convert from or can convert to.	The color space metadata must either be correct, or you must be prepared to remove it from the output.

Color space versus video resolution

Color space refers to the range of color. MediaLive supports the following color spaces:

- SDR (standard dynamic range)
- HDR (high dynamic range) color spaces

Resolution refers to the video pixel count. MediaLive supports the following resolutions:

- SD (standard definition).
- HD (high definition).
- UHD (ultra-high definition). For UHD, MediaLive resolutions up to 4K.

The following combinations of color space and resolution are typically used:

- SDR color space can be associated with SD, HD, and UHD video.
- HDR color space can be associated with HD or UHD video.

An HDR color space isn't typically associated with SD content, but MediaLive *does* support ingesting this combination.

General information about color space

Following is some general information about color space.

Topics

Components of color space

There are four components to color space:

- The specific *color space* that applies to the video content. The color space specifies a range of pixel colors that can apply to the content.
- The *color space metadata*, which identifies the color space being used. If this metadata is present, the content is said to be *marked* for a color space.
- The *brightness function* that applies to the color space. The brightness function controls the brightness of each pixel. The brightness is also known as gamma tables, electro-optical transfer function (EOTF), and transfer function.

- The *brightness metadata*, which identifies the brightness function being used.
- The *display metadata* that applies to the color space. Not all standards have this metadata.

The video might use a specific *color space* and a specific *brightness function*. The video might also carry *color space metadata* that describes aspects of the color.

Color space standards that MediaLive supports

Each color space standard follows a specific standard for the color space, and specific standards for the three sets of color data.

To read this table, find a color space in the first column, then read across to identify the standards for the color space and the three sets of color data.

MediaLive term for the color space	Complies with this color space standard	Complies with this brightness function (gamma) standard	Complies with this standard for display metadata
Rec. 601	Rec. 601	BT.1886	Not applicable. This color space doesn't include display metadata.
Rec. 709	Rec. 709	BT.1886	Not applicable. This color space doesn't include display metadata.
HDR10	Rec. 2020	SMPTE ST 2084 (PQ)	SMPTE ST 2086
HLG or HLG 2020	Rec. 2020	HLG rec. 2020 (ARIB_STD-B67/HLG)	Not applicable. This color space doesn't include display metadata.
Dolby Vision 8.1	Rec. 2020	SMPTE ST 2084 (PQ)	Proprietary Dolby Vision 8.1 metadata (RPU), on a per-frame basis, and SMPTE ST 2086 on a per-stream basis.

Default behavior

The default behavior for a channel is to pass through the color space and pass through the uncorrected color space metadata. Therefore, if you want to pass through the color space to all the outputs, you can stop reading this entire section about handling color space.

Passing through the color space

You can set up to *pass through* the color space. These rules apply:

- The color space can be a supported color space or an unsupported color space.
- You can set up to include or remove the color space metadata.
- You can set up to pass through the color space in some outputs and to convert it in other outputs.

Converting the color space

You can set up to *convert* the color space itself, to change the pixels in the video.

MediaLive can perform specific color space conversions.

You can set up each output in the channel for different handling. For example, you can set up one output to convert the color space to HDR10, set up one output to convert to HLG, and set up another output to pass through the color space.

Topics

Supported types of conversion

In the following table, find an output color space in the first column. Then read the second column for the source color spaces that can produce that output. The other columns specify whether MediaLive can perform the conversion using the traditional color space mapping or using a 3D LUT file.

Any of these color spaces in the source	Desired color space in the output	MediaLive supports this conversion with the standard mechanism	MediaLive supports this conversion using a 3D LUT file
Rec. 709, HLG, HDR10	Rec. 601	Yes	Yes

Any of these color spaces in the source	Desired color space in the output	MediaLive supports this conversion with the standard mechanism	MediaLive supports this conversion using a 3D LUT file
Rec. 601, HLG, HDR10	Rec. 709	Yes	Yes
Rec. 601, Rec. 709, HLG	HDR10	Yes	Yes
Rec. 601, Rec. 709, HDR10	HLG	No	No
HDR10	Dolby Vision 8.1	Yes	No
Dolby Vision 8.1	Any color space supported by MediaLive	No	No

Support for processing with 3D LUT files

You can configure a channel to use a 3D LUT file for conversion. Or you can use the standard MediaLive mechanism for conversion.

You provide a list of 3D LUT files. Each 3D LUT file contains color mapping information for a specific source/output combination. For example, one file contains information for converting Rec. 709 to HDR10.

Using 3D LUT files

These rules apply to using 3D LUT files:

- **Sourcing of 3D LUT files.** You must provide the 3D LUT files. MediaLive doesn't have built-in files.
- **One file for each combination.** You must provide a file for each source/output combination. For example, a file for converting Rec. 601 to HDR10.
- **Maximum 8 files.** You can provide a maximum of 8 files for each channel. This means that MediaLive supports up to 8 source/output conversion combinations.

- **Global application.** MediaLive uses a specific file in all the outputs where that file applies. For example, if there is a file to convert Rec. 601 to HDR10, MediaLive uses that file in every output that it applies to. You can't configure some outputs to use the standard mechanism for conversion.

Contents of the 3D LUT files

The following rules apply to the contents of the files:

- **Format.** You must make sure that each 3D LUT file follows the .cube 3D LUT format.

- **Maximum one file per combination.** You can provide only one 3D LUT file for each combination. You can't configure some outputs to use a different 3D LUT file. When MediaLive reads the list of 3D LUT files, it uses the first file that it finds for a source/output combination.
- **HDR10 luminance.** MediaLive supports conversion of HDR10 content with a maximum luminance of 1000 nits to 4000 nits, but it only supports one maximum luminance. When MediaLive reads the list of 3D LUT files, it finds the first file for each conversion from HDR10. Even if you one file for 1000 nits and one for 4000 nits (for example), MediaLive uses only the first file it encounters. Therefore the following guidelines apply:
  - You should make sure that all the HDR10 content in all the inputs in one channel have the same maximum luminance. If a source has a different maximum luminance, MediaLive will convert the content, but the outputs will have sub-optimal luminance.
  - In each 3D LUT file for converting from HDR10, make sure that the luminance handling is appropriate for the luminance of the source.

Input requirements

MediaLive can work with the supported color space in all supported types of input with the following notes.

Elemental Link input

MediaLive can't read the color space metadata in a source from an AWS Elemental Link device. The workaround when you set up the input is to specify the color space that applies, as described in the section called “Step 1: Configure the inputs”.

Source when converting to Dolby Vision 8.1

- The video source must be HD or 4K resolution. In other words, the source must be 1080p or higher.
- The video source must be HDR10. If MediaLive encounters a portion of non-HDR10 content, it passes through the color space and color space metadata for that portion.
- The video source can't be a file. This means that the source can't be a VOD asset in an MP4 file or in a transport stream file.

These constraints are stipulated by Dolby Vision 8.1, and relate to the minimal video quality required to produce Dolby Vision 8.1 outputs that meet the Dolby Vision 8.1 standard.

Output requirements

Supported output types

All color space types except Dolby Vision 8.1 can be set up in all output group types.

Dolby Vision 8.1 can be set up only in the following output group types:

- Archive
- CMAF Ingest
- HLS
- UDP

Supported output codecs

The following table specifies the supported video codecs for the output color spaces.

Output color space	AVC (H.264)	HEVC (H.265)
Rec. 601	Yes	Yes
Rec. 709	Yes	Yes
HDR10		Yes
HLG		Yes
Dolby Vision 8.1		Yes

Supported video profile for HDR10 or Dolby Vision 8.1 outputs

For HDR10 or Dolby Vision 8.1 outputs, the video profile must include the term *10BIT*.

Handling of metadata when converting

When you set up to convert the color space, you can set up to include or omit the color space metadata.

- Include color space metadata. MediaLive will convert the color space metadata to accurately describe the new color space.
- Omit the color space metadata. You might want to remove the color space metadata because the downstream system can't handle it properly.

  When MediaLive removes the metadata, the source still has a color space but it doesn't have information that identifies the color space. Removing the metadata doesn't necessarily degrade the color. Removing it might only mean that the downstream player can't implement enhancements to make the color even richer.

Configuring the inputs

> **Note**
>
> This section assumes that you are familiar with creating or editing a channel, as described in *Setup: Creating a channel* .

This section describes how to set up each of the sources (inputs). It applies both when you pass through the color space to the output and when you convert the color space.

Follow these steps for each input in the channel.

1. Make sure that your inputs comply with the requirements.
2. On the **Create channel** page, in the **Input attachment** section for the input, in the **General input settings** section, group, open the **Video selector** field.
3. Set these fields:

   - **Color space**: Choose **Follow**.

(The other options apply only to complicated color space situations.)

- **Color space usage**: Leave the default. This field is ignored when you set **Color space** to **Follow**.

This combination of values indicates that the color space metadata in the content correctly identifies the color space, therefore MediaLive can use that metadata.

4. Obtain the values for the Max CLL and Max FALL for the content, but only if the following situation applies:

   - The input is for a MediaLive device such as AWS Elemental Link.
   - The input color space is HDR10. (This means that the output from the Link device is HDR10.)
   - You plan to pass through the color space to the output.

   You need this information because MediaLive can't read the metadata from an AWS Elemental Link device. Instead, you will be able to enter the color space and the display metadata (Max CLL and Max FALL) manually, in the next step.

   You don't need these values if you plan to convert this input from HDR10 to another color space.

5. Complete **Color space settings** as follows:

   - If the situation in step 4 applies, choose **HDR10** (to identify the source color space). Then, if you obtained metadata values, enter them in the **Max CLL** and **Max Fall** fields that appear (to provide the metadata that is missing from the input).

   - If the situation doesn't apply, choose **Don't include**.

Configuring color space handling in each output

> **Note**
>
> This section assumes that you are familiar with creating or editing a channel, as described in *Setup: Creating a channel* .

This section describes how to set up each output in the channel for the desired color space handling — passthrough or conversion. You can set up each output in a different way.

Follow these steps for each output in the channel.

Topics

Configure enhanced VQ mode

You must decide if you should enable enhanced VQ mode in the output. This mode applies only to outputs that use H.264.

1. In the following table, find the planned handling in the first column, then read across to identify the action to take.
2. To enable enhanced VQ mode, see the section called “Video – enhanced VQ”.

Planned handling	Details	Action
Pass through	You already plan to enable enhanced VQ to improve the video quality.	Enable the mode.
Convert to SDR	The inputs contain both SDR and HDR content.	You must enable the mode.
Convert to SDR	The inputs contain only SDR content. For example, all the inputs are Rec. 709, and you want to convert the content to Rec. 601.	Enabling the mode is optional.

Planned handling	Details	Action
Any conversion	There is no HDR10 or HLG in any of the inputs.	Enabling the mode is optional.
Any conversion	You already plan to enable enhanced VQ to improve the video quality.	Enable the mode.

Prepare 3D LUT files

Perform these steps if you plan to convert the color space in an output, and you want to use 3D LUT files for that conversion. You must obtain the 3D LUT files and store them in an accessible location.

1. Read the section called "3D LUT files" for rules and requirements.
2. Identify the conversion combinations that you want to perform across all outputs in the channel. You can perform a maximum of 8 conversion combinations, because you can specify a maximum of 8 files.
3. Obtain the 3D LUT file for each combination. For files for HDR sources, make sure that you obtain the file that matches the maximum luminance of the source.

   Make sure that you know which source/output combination each 3D LUT file handles. There is no metadata in the files that lets MediaLive discover the combination. Instead, when MediaLive needs to choose a file, it reads the information that you specify when you configure the channel (in the next step).
4. Store the files in one or more buckets in Amazon S3. Make a note of the URLs for the files you want to use in this channel. Make sure that MediaLive has access to the Amazon S3 bucket or buckets.

Example

For example, you might have sources in Rec. 601, Rec. 709, and HDR10. You might want to convert the HDR10 sources to Rec. 709 in one output, and you might want to convert the SDR sources to HDR10 in another output. The following table summarizes your intentions.

Source	Rec. 709 output	HDR10 output
Rec. 601	Convert using standard mechanism	Convert using 3D LUT file
Rec. 709	No conversion	Convert using 3D LUT file
HDR10	Convert using 3D LUT file	No conversion

Therefore you need to provide three files:

- Rec. 601 to HDR10
- Rec. 709 to HDR10
- HDR10 to Rec. 709

Note that when there is no conversion, you shouldn't provide a file. MediaLive doesn't support handling that retains the source color space and only changes the color mapping. MediaLive would never use that file.

Note also that you decided not to provide a file to convert Rec. 601 to Rec. 709 because the difference in the output is minimal. MediaLive will convert the Rec. 601 source to Rec. 709, but it will do so using the standard mechanism.

Configure the channel to use 3D LUT files

Perform these steps if you plan to convert the color space in an output, and you want to use 3D LUT files for that conversion. You must configure MediaLive to use those files.

1. On the **Create channel** page, in the **General settings** section, select **Color Correction Settings**. Slide the **Enable Color correction settings** field.
2. Choose **Add global color corrections** up to 8 times. In each line, enter the following information:
   - The URL for the 3D LUT file.
   - The input (source) color space that this file handles.
   - The output color space that this file handles.

Runtime handling

When you start the channel, MediaLive will use the 3D LUT files as follows:

- MediaLive reads the color space metadata in each source video frame. For example, for one frame, the metadata might specify that the color space is Rec. 601.
- It reads the color space you have configured for each output. For example, the video output might be set up for conversion to HDR10.
- MediaLive reads the information you configured about each file and finds the first file that matches the source and output.

  If the source and output are the same (for example, Rec. 601 source and Rec. 601 output), MediaLive doesn't look for a file. It always passes through the color space.

  If there is more than one 3D LUT file with the same input and output specified, MediaLive uses the first file it encounters.
- MediaLive uses that file to convert that combination of source and output.
- If there is no file for this combination, MediaLive performs the conversion using the standard mechanism (that is, without using 3D LUT color mapping).

Setting up for passthrough

You can set up to pass through the source color space in one or more outputs. The key fields to set are **Color space** and **Color metadata**.

1. On the **Create channel** page, in the **Output groups** section, choose the output that contains the video. Display the **Stream settings** section, and choose the **Video** section.
2. For **Codec settings**, choose a codec. For information about the color spaces that each codec supports, see the section called “Output requirements”.
3. Choose **Codec details**. More fields appear. Choose **Additional settings**. More fields appear.

   In **Color metadata**, choose **Insert** or **Ignore** to specify how you want to handle the color space metadata.
4. Choose **Color space**. The **Color space settings** field appears.

   Choose **Color space passthrough**. (Or choose **Don't include**, which is equivalent to **Color space passthrough**.)

Setting up to convert

You can set up to convert the color space in one or more outputs. You must set up several fields, each in a specific way.

1. On the **Create channel** page, in the **Output groups** section, select the output that contains the video. Display the **Stream settings** section, and choose the **Video** section.
2. Complete the **Width** and **Height** fields to specify a valid resolution. Make a note of whether you are specifying an SD, an HD, or a UHD resolution.
3. In **Codec settings**, choose a codec. For information about the color spaces that each codec supports, see the section called “Output requirements”.
4. Choose **Codec details**. More fields appear. Set the **Profile**, **Tier** (for H.265 only), and **Level** fields:
   - If the resolution is SD, enter values that suit your requirements.
   - If the resolution is an HD or UHD resolution, set the tier and level to suit your requirements, and set the profile as follows:
     - If the output color space will be an HDR color space, you must choose one of the profiles that has **10BIT** in the name.
     - If the output color space will be an SDR color space, you can choose any profile.
5. Choose **Color space**. The **Color space settings** field appears.

   Set the field to the color space to convert to.

   If you choose **HDR10**, the **Max CLL** and **Max FALL** fields appear. Complete these fields to set the display metadata for the HDR10 output video.
6. Go back to **Codec details** and choose **Additional settings**. More fields appear, including **Color metadata**.

   In **Color metadata**, choose **Insert** or **Ignore** to specify how you want to handle the color space metadata. Typically, you omit it only if you know that the downstream system can't handle it properly.

Results for different color space handling

This section describes how MediaLive handles the color space and color space metadata that it encounters in the source input, depending on how you set up the color space in the output.

Topics

General process for handling color space conversion

In an output that specifies to convert the color space, MediaLive performs the following steps on each video frame in the output.

Initial verifications

- MediaLive verifies that the source video meets the input requirements and that it is in a supported color space. If it fails this verification, MediaLive always passes through the color space.
- If the source video does meet the requirements, MediaLive verifies that you have correctly configured the output video and codec. If it fails this verification, MediaLive uses the specified codec, but it passes through the color space.
- MediaLive determines if the channel has been configured to use 3D LUT files.

Handling with 3D LUT files configured

MediaLive looks at the source and output color space for each frame.

- If the source and color space are identical, MediaLive doesn't change the color space and therefore doesn't look for a 3D LUT file. For example, if the source is HDR10 and you set up an output for HDR10, MediaLive leaves the color space as it is in the source.
- If the source and color space are different, MediaLive looks for a file that corresponds to the source/output color space combination:
  - If it finds a file, it uses that file for conversion.
  - If it doesn't find a file, it converts the color space using the standard mechanism.

Handling without 3D LUT files configured

If the channel hasn't been configured to use 3D LUT files, then MediaLive converts the color space using the standard mechanism.

For detailed information about the results of each kind of source/output conversion, see the sections that follow.

Result when passing through color space

Read this section if you set up one or more outputs to pass through the color space. The following table shows how MediaLive handles each type of color space that it encounters in the source.

Color space that MediaLive encounters	How MediaLive handles the color space
Content in any color space that MediaLive supports	Doesn't touch the color space or brightness in the output. Passes through any of the three color format metadata fields that are present.

Result when converting color space to SDR

Read this section if you set up one or more outputs to convert the color space to Rec. 601 or Rec. 709. The following table shows how MediaLive handles each type of color space that it encounters in the source.

Color space that MediaLive encounters	How MediaLive handles the color space
Content in the same SDR color space	• Doesn't touch the color space in the output. • Passes through the color space metadata. • Passes through the brightness metadata.
Content in the other SDR color space	• Converts the content to the chosen SDR color space and brightness function. The conversion maps the pixels to code values that represent the same color as the original code values. • Changes the color space metadata to specify the new color space.

Color space that MediaLive encounters	How MediaLive handles the color space
	• Passes through the brightness metadata. This is appropriate because the two SDR color spaces use the same brightness function.
Content in HDR10	When the output codec is H.264 and if you have enabled enhanced VQ, MediaLive does the following: • Converts the content to the chosen SDR color space and brightness function. The conversion fits the colors into the smaller color space. • Changes the color space metadata to specify the new color space. • Changes any brightness metadata to specify the new standard. • Removes any display metadata. After the conversion, the content complies completely with the new color space. The color will be less rich. The color will match the new brightness function. If you haven't enabled enhanced VQ, MediaLive doesn't convert anything. It passes through the color space metadata, any brightness metadata, and any display metadata.

Color space that MediaLive encounters	How MediaLive handles the color space
	When the output codec is H.265, MediaLive does the following: • Converts the content to the chosen SDR color space and brightness function. The conversion fits the colors into the smaller color space. • Changes the color space metadata to specify the new color space. • Changes any brightness metadata to specify the new standard. • Removes any display metadata. After the conversion, the content complies completely with the new color space. The color will be less rich. The color will match the new brightness function.

Color space that MediaLive encounters	How MediaLive handles the color space
Content in HLG	When the output codec is H.264 and if you have enabled enhanced VQ, MediaLive does the following: • Converts the content to the chosen SDR color space and brightness function. The conversion fits the colors into the smaller color space. • Changes the color space metadata to specify the new color space. • Changes any brightness metadata to specify the new standard. After the conversion, the content complies completely with the new color space. The color will be less rich. The color will match the new brightness function. If you haven't enabled enhanced VQ, MediaLive doesn't convert anything. It passes through the color space metadata, and any brightness metadata.

Color space that MediaLive encounters	How MediaLive handles the color space
	When the output codec is H.265, MediaLive does the following: • Converts the content to the chosen SDR color space and brightness function. The conversion fits the colors into the smaller color space. • Changes the color space metadata to specify the new color space. • Changes any brightness metadata to specify the new standard. After the conversion, the content complies completely with the new color space. The color will be less rich. The color will match the new brightness function.

Result when converting color space to HDR10

Read this section if you set up one or more outputs to convert the color space to HDR10. The following table shows how MediaLive handles each type of color space that it encounters in the source.

Color space that MediaLive encounters	How MediaLive handles the color space
Content in an SDR color space	• Converts the content to the new color space and brightness function. The conversio n maps the pixels to code values that represent the same color as the original code values. • Changes the color space metadata to specify the new color space.

Color space that MediaLive encounters	How MediaLive handles the color space
	• Changes the brightness metadata to specify the new standard. • If you completed **Max CLL** and **Max FALL**, the values in those fields are inserted in the display metadata. There is no change to the pixel values of the content. In effect, this conversion fits the smaller SDR color space into the larger HDR color space and maps the pixels to new code values that represent the same color. The conversion doesn't actually make the existing color richer. However, the bright parts of the content are brighter, and the dark parts are darker.
Content in HDR10	• Doesn't touch the color space in the output. • Passes through the color space metadata. • Passes through the brightness metadata and the display metadata.

Color space that MediaLive encounters	How MediaLive handles the color space
Content in HLG	• There is no color space conversion—there is no change to the pixel values of the content. This is appropriate because HDR10 and HLG use the same color space (they are different only in brightness function and display metadata). • Converts the content to the new brightness function. • Changes the color space metadata to the new color space. • Change the brightness metadata to specify the new standard. • If you completed **Max CLL** and **Max FALL**, MediaLive inserts the values from those fields into the display metadata.

Result when converting color space to Dolby Vision 8.1

Read this section if you set up one or more outputs to convert the color space to Dolby Vision 8.1. The following table shows how MediaLive handles each type of color space that it encounters in the source.

Color space that MediaLive encounters	How MediaLive handles the color space
Content in HDR10	When you convert suitable content to Dolby Vision 8.1, MediaLive makes the following changes: • It doesn't change the pixel values, because HDR10 and Dolby Vision 8.1 both use the same color space. • It changes the color space metadata to identify the new color space.

Color space that MediaLive encounters	How MediaLive handles the color space
	• It applies the new brightness function to the content. • It calculates the Dolby Vision 8.1 display metadata for the content. After the conversion, the color space hasn't changed. However, the bright parts of the content are brighter, and the dark parts are darker.
Content in any other supported color space	MediaLive passes through the color space and color space metadata for that portion,

Reference: Location of fields

Read this section if you know how to handle color space in MediaLive, and you only need a reminder of where the fields are located in the MediaLive Console. The information is sorted by the location of the fields on the **Channel** page, from top to bottom.

Topic	Location on the Channel page		Field
Input handling	**Input attachments**	**Video Selector**	**Color space**
			Color space usage
Enter the display metadata for an input from a AWS Elemental Link device	**Input attachments**	**Video Selector**, then **Color space settings**	**Max CLL**
			Max Fall
Configure the channel to use 3D LUT files	**General settings**	**Color correction settings**	**Url**
			Input color space
			Output color space

Topic	Location on the Channel page		Field
Output, configure the video codec	**Output groups**, then **Outputs**	**Stream settings**, then **Video**	**Codec settings**
		Stream settings, then **Video**, then **Codec settings**, then **Codec details**	**Profile**
			Tier
			Level
Output, convert the color space	**Output groups**, then **Outputs**	**Stream settings**, then **Video**, then **Color space**	**Color space settings**
Output, include or omit color space metadata	**Output groups**, then **Outputs**	**Stream settings**, then **Video**, then **Codec settings**, then **Codec details**, then **Additional settings**	**Color metadata**
Output, specify display metadata to include, only if you are converting to HDR10	**Output groups**, then **Outputs**	**Stream settings**, then **Video**, then **Color space**, then **Color space settings**	**Max CLL**
			Max Fall
Output, set up enhanced VQ, only if the output codec is H.264	**Output groups**, then **Outputs**	**Stream settings**, then **Video**, then **Codec settings**, then **Additional encoding settings**	**Quality level**
			Filter settings

Handling complex color space conversions

> **Important**
>
> Read the section called "Which section to read" to determine if you should read this section.

All video belongs to a specific color space. The color space defines the range of color for the video. Video can include color space metadata. This metadata provides information about the color space. When the color space metadata is missing, the video still has a color space, but it is impossible for a video processor such as MediaLive to manipulate the color space.

You can control how MediaLive takes the color space and color space metadata in a video source and manipulates it in the video output. You can set up each output video encode to convert or pass through the color space, and to include or omit the color space metadata.

Default behavior

The default behavior is to pass through the color space and pass through the color space metadata.

Topics

Which section to read

There are two sections in this guide about handling color space — this section, and the section the section called "Video – color space conversion".

Read the requirements in the section called "Determine if this section applies to your channel" in the section called "Video – color space conversion". If your content doesn't meet all those requirements, then follow the procedures in this section instead.

This section provides procedures for dealing with complicated situations that include the following:

- Assessing the accuracy of the color space metadata in content.
- Cleaning up the metadata in content.

- Content that is one input, where the color space switches within the input.
- Converting content that is a combination of supported and unsupported color spaces.

These requirements often apply to content that is a VOD file that you are converting to a live stream. The VOD file might have been created by stitching together several different sources, each with a different color space. It might contain older content with an unknown color space and/or with missing or inaccurate metadata.

Options for handling color space

All video belongs to a specific color space. The color space defines the range of color for the video. Video can include color space metadata. This metadata provides information about the color space. When the color space metadata is missing, the video still has a color space, but it is impossible for a video processor such as MediaLive to manipulate the color space.

You can control how MediaLive takes the color space and color space metadata in a video source and manipulates it in the video output.

You can set up each output video encode to handle the color space in different ways:

Option	Handling of color space	Handling of color space metadata
Pass through and include	Pass through	Pass through (corrected or original)
Pass through and remove	Pass through	Remove

Option	Handling of color space	Handling of color space metadata
Convert and include	Convert	Generate new color space metadata
Convert and remove	Convert	Remove

Scope of handling in outputs

You can set up each output in the channel for different handling. For example, you can set up one output to convert the color space to HDR10, set up one output to convert to HLG, and set up another output to pass through the color space. For more information, see the section called "Passthrough" and the section called "Conversion" .

General information about color space

Following is some general information about color space.

Topics

Definition

There are four components to color space:

- The specific *color space* that applies to the video content. The color space specifies a range of pixel colors that can apply to the content.
- The *color space metadata*, which identifies the color space being used. If this metadata is present, the content is said to be *marked* for a color space.

- The *brightness function* that applies to the color space. The brightness function controls the brightness of each pixel. The brightness is also known as gamma tables, electro-optical transfer function (EOTF), and transfer function.

- The *brightness metadata*, which identifies the brightness function being used.
- The *display metadata* that applies to the color space. Not all standards have this metadata.

The video might use a specific *color space* and a specific *brightness function*. The video might also carry *color space metadata* that describes aspects of the color.

Supported color space standards

Each color space standard follows a specific standard for the color space, and specific standards for the three sets of color data.

To read this table, find a color space in the first column, then read across to identify the standards for the color space and the three sets of color data.

MediaLive term for the color space	Complies with this color space standard	Complies with this brightness function standard	Complies with this standard for display metadata
Rec. 601 or Rec. 601	Rec. 601	BT.1886	Not applicable. This color space doesn't include display metadata.
Rec. 709 or Rec. 709	Rec. 709	BT.1886	Not applicable. This color space doesn't include display metadata.
HDR10	Rec. 2020	SMPTE ST 2084 (PQ)	SMPTE ST 2086
HLG or HLG 2020	Rec. 2020	HLG rec. 2020	Not applicable. This color space doesn't include display metadata.

MediaLive term for the color space	Complies with this color space standard	Complies with this brightness function standard	Complies with this standard for display metadata
Dolby Vision 8.1	Rec. 2020	SMPTE ST 2084 (PQ)	Proprietary Dolby Vision 8.1 metadata (RPU), on a per-frame basis, and SMPTE ST 2086 on a per-stream basis.

Passing through the color space

You can set up to *pass through* the color space. You can set up to include or remove the color space metadata. For passthrough to produce the desired quality in the video output, the color space metadata must be accurate.

Here are the possible combinations for passthrough:

- Pass through the color space, pass through the color space metadata without correcting it (because you know that it is accurate).
- Pass through the color space, pass through the color space metadata after correcting it.
- Pass through the color space, remove the color space metadata without correcting it. You might want to remove the color space metadata because the downstream system can't handle it properly.

  When MediaLive removes the metadata, the source still has a color space but it doesn't have information that identifies the color space. Removing the metadata doesn't necessarily degrade the color. Removing it might only mean that the downstream player can't implement enhancements to make the color even richer.

Default behavior

The default behavior is to pass through the color space and pass through the uncorrected color space metadata.

Converting the color space

You can set up to *convert* the color space itself—to change the pixels in the video.

MediaLive can convert only color spaces that it supports. See the section called “Supported color spaces”.

Here are the possible combinations for conversion:

- Convert the color space, and include color space metadata. MediaLive will convert the color space metadata to accurately describe the new color space.
- Convert the color space, but omit the color space metadata. You might want to remove the color space metadata because the downstream system can't handle it properly.

  When MediaLive removes the metadata, the source still has a color space but it doesn't have information that identifies the color space. Removing the metadata doesn't necessarily degrade the color. Removing it might only mean that the downstream player can't implement enhancements to make the color even richer.

> **Warning**
>
> If the content provider can't tell you what color space applies to the input, you shouldn't try to convert the color space. Doing so might degrade the video quality. You should pass through the color space. You should also remove the color space metadata, so that the downstream system doesn't read information that might be inaccurate.

MediaLive converts from one color space to another based on the metadata in the source content. MediaLive doesn't examine the video to try to determine whether it actually matches the color space identified in the metadata.

Supported types of conversion

The following table identifies the color spaces in the source that can be converted to a specific color space in the output.

Any of these color spaces in the source	Can be converted to this color space in the output
Rec. 709, HLG, HDR10	Rec. 601
Rec. 601, HLG, HDR10	Rec. 709
Rec. 601, Rec. 709, HLG	HDR10
None. Conversion to HLG isn't supported	HLG
HDR10 If MediaLive encounters a portion of non-HDR10 content, it passes through the color space and color space metadata for that portion,	Dolby Vision 8.1

Input and output requirements

Topics

- Supported inputs
- Supported output types
- Supported output codecs

Supported inputs

AWS Elemental Live can work with the supported color space in all supported types of input with the following notes:

- Handling Elemental Link inputs: MediaLive can't read the color space metadata in a source from an AWS Elemental Link device. The workaround when you set up the input is to specify the color space that applies, as described in the section called “Correct metadata with force”.
- Converting to Dolby Vision 8.1:

- The video source must be HD or 4K resolution. In other words, the source must be 1080p or better.
- The video source must be HDR10. If MediaLive encounters a portion of non-HDR10 content, it passes through the color space and color space metadata for that portion,
- The video source can't be a file. This means that the source can't be a VOD asset in an MP4 file or a VOD asset in a transport stream.

These constraints are stipulated by Dolby Vision 8.1, and relate to the minimal video quality required to produce Dolby Vision 8.1 outputs that meet the Dolby Vision 8.1 standard.

Supported output types

All color space types except Dolby Vision 8.1 can be set up in all output group types.

Dolby Vision 8.1 can be set up only in the following output group types:

- Archive
- CMAF Ingest
- HLS
- UDP

Supported output codecs

The following table specifies the supported codecs for the output color spaces.

Output color space	AVC (H.264)	HEVC (H.265)
Rec. 601	Yes	Yes
Rec. 709	Yes	Yes
HDR10		Yes
HLG		Yes
Dolby Vision 8.1		Yes

Supported video profile for HDR10 or Dolby Vision 8.1 outputs

For HDR10 or Dolby Vision 8.1 outputs, the video profile must include the term *10BIT*.

General procedure for handling color space

The procedure for handling color space in the channel is the same for both passing through and converting the color space in the outputs.

1. You must assess the color space in all the inputs, and determine if you can handle the color space according to your preference. See the section called “Assess the color spaces in the sources”.
2. You must assess the source to make sure the color space metadata is correct.
   - For passthrough: If you plan to include the color space metadata, you must assess it. If the metadata isn't correct, downstream players won't handle the color space correctly.
   - For conversion: MediaLive reads this metadata to determine the color space of the source, so that it can apply the correct conversion formula. Therefore, even if you plan to remove the metadata in the outputs, you must assess the metadata.

   See the section called “Step 1: Assess the metadata”.
3. If you need to correct the color space metadata, you do so in the input. You configure each input separately.

   See the section called “Step 2: Correcting metadata”.
4. Set up the output to pass through or convert the color space, and to include or omit the color space metadata. See the section called “Configuring outputs”.

Assess the color spaces in the sources

1. Speak to the content provider of each input. Obtain the following information:
   - The names of the color spaces that apply to the content.
   - Whether each input consists of only one color space or several color spaces.
   - Whether the color space metadata is accurate. (You will use this information in the next section.)
2. Read the following information to determine if there are reasons not to pass through or not to convert the color space.

Topics

Unknown color space

If the content provider can't tell you what color space applies to the input, you shouldn't try to convert the color space. Doing so might degrade the video quality.

You might be able to pass through the color space. In this case, you should remove the color space metadata, so that the downstream system doesn't read information that might be inaccurate.

Restrictions on passthrough

Passthrough of supported color spaces

MediaLive can pass through color spaces that it supports.

Passthrough of unsupported color spaces

MediaLive might be able to pass through color spaces that it doesn't support. Any of the following might apply:

- MediaLive might be able to ingest the input, and to pass through the color space and the color space metadata.
- Or it might ingest the input but produce unacceptable output.
- Or it might fail to ingest the input, so that the event follows the input loss behavior routine (for example, it might display a slate in the output).

Passthrough and the output codec

Even if MediaLive supports the color space that you want to pass through, there might be a restriction because of the output codec.

If you want to pass through the color space in even one output, then every input in the channel must be in color spaces that are supported by the codec for the output. For information about codecs, see the section called “Supported output codecs”.

For example, you have an output where you want to pass through the color space. You want to encode that output with H.264. Assume that one of the channel inputs includes content in Dolby Vision 8.1. However, the Dolby Vision color space (from the input) can't be included in H.264. MediaLive will accept the configuration, but the portions of the output in the unsupported color space will be degraded.

The workaround is to choose an output codec that is supported by all the color spaces in all the inputs.

Note how the rule for passthrough in a channel is based on the color space for all the inputs.

Restrictions on conversion

Even if MediaLive supports conversion to a specific color space, there might be a restriction because of the output codec.

If you want to convert to a specific color space in an output, then the codec you set in that output must support that color space.

For example, you have an output where you want to encode with H.264, and you want to convert all the source color spaces to HDR10. However, HDR10 can't be included in H.264. MediaLive won't let you configure in this way. When you choose H.264, the option for HDR10 is removed from the field where you specified the output color space.

The workaround is to choose an output codec (H.265) that is supported with the color space conversion.

Note how the rule for conversion in a channel is based on the color space and codec for the individual output.

Handling color space metadata in the inputs

You must assess the color space of the inputs and determine if the color space metadata needs to be cleaned up. You must perform this assessment in order to decide about the correct handling for the outputs.

Topics

- Step 3: Set up inputs to correct metadata

Step 1: Assess the color space metadata in the sources

Before you can set up the outputs, you must determine if you need to modify the color space metadata in the inputs. To make this decision, you must assess the quality of the metadata in the inputs.

> **Important**
>
> The handling on the input side of the event is about changing the color space metadata, not changing the color space itself. The handling is about changing the metadata to correctly identify the color space in the input, in preparation for the planned handling in the outputs.
> The conversion of the video to a different color space occurs in the section called "Configuring outputs".

To assess the inputs

1. You should have already obtained information about the accuracy of the color space metadata in all the inputs.
2. Make a note of the presence and accuracy of the metadata for all the color spaces in all the inputs.

   The color space metadata is accurate if the following applies:

   - It is present in the input and it accurately identifies the color space, which means that the color space is accurately marked.

   The color space metadata might be present, but it might be inaccurate in one or more ways:

   - Incorrect: The metadata doesn't match the color space.
   - Unknown: The metadata marks the color space as *unknown*.
   - Not supported: The metadata specifies a color space that MediaLive doesn't support. MediaLive doesn't read this metadata.
   - Missing: All or part of the video might not have color space metadata.

3. This step applies only for the following situation:

   - The input is for a MediaLive device such as AWS Elemental Link.
   - The input color space is HDR10.
   - You plan to pass through the color space to the output.

   Obtain the values for the Max CLL and Max FALL for the content.

   MediaLive can't read the metadata from an AWS Elemental Link device. But you will be able to enter the color space and the display metadata (Max CLL and Max FALL) manually, in the channel configuration.

   You don't need these values if you plan to convert this input from HDR10 to another color space.

Step 2: Options for correcting metadata

In step 1, you assessed the status of the color space metadata in the inputs. You must now decide if you can clean up any inaccurate metadata.

MediaLive can clean up the color space metadata for any color space except Dolby Vision 8.1 or an unsupported color space.

> **Note**
>
> If you want to convert the color space in your channel, the metadata for all the inputs must be either accurate or cleaned up. If there is even one input that you can't clean up, you won't be able to convert the color space in the outputs. You will have to set up to pass through the color space.
> If you want to pass through the color space and include its metadata, the metadata for all the inputs must be either accurate or cleaned up. The downstream system reads this metadata, so it must be accurate. If there is even one input that you can't clean up, you can pass through the color space, but you should omit the color space in the output.

Topics

- Scenario B – Metadata can be corrected with force
- Scenario C – Correct the metadata with fallback
- Scenario D – Metadata can't be corrected

Scenario A – Metadata is accurate

During assessment of the input, you might have determined the following:

- The content is in one color space, the color space is supported, and the color space metadata is accurate.
- Or different portions of the content are in different color spaces, and the color space metadata is accurate for each portion.

You have these options for handling the metadata in the output:

Include the metadata

Follow the procedure in the section called “Step 3: Set up inputs”, and set the key fields as follows:

- **Color space** field – Set to **FOLLOW**
- **Color space usage** field – MediaLive ignores this field.

During processing, MediaLive will read the metadata, in order to identify the color space.

Remove the metadata

You might have already decided to remove the color space metadata even though it is accurate. For example, the color space might change frequently within the input, or between one input and another. You know that there is a system downstream of MediaLive that can't handle changes in the metadata.

You can still convert or pass through the color space. It is safe to convert the color space because the metadata is reliable.

Follow the procedure in the section called “Step 3: Set up inputs”, and set the key fields as follows:

- **Color space** field – Set to **FOLLOW**
- **Color space usage** field – MediaLive ignores this field.

During processing, MediaLive will read the metadata, in order to identify the color space.

Scenario B – Metadata can be corrected with force

During assessment of the input, you might have determined the following:

- The content is in one color space, and that is a supported color space.
- The color space metadata is inaccurate. It could be any combination of inaccurate, missing, unknown, or unsupported (inaccurately marked as a color space that MediaLive doesn't support).

Note that this is the scenario that always applies if the input is from an AWS Elemental Link device.

You have this option for handling the metadata in the output:

Correct the metadata

You can correct the metadata. Follow the procedure in the section called “Step 3: Set up inputs”, and set the key fields as follows:

- **Color space** field – Set to the color space that has unacceptable metadata.
- **Color space usage** field – Set to **FORCE**

During processing, MediaLive will create metadata of the specified color space for all missing, unmarked, and unknown metadata. It will also change all existing metadata to the specified color space. (It will *force* the metadata.)

After ingest, all the content in the input will be consistently marked as one color space.

Scenario C – Correct the metadata with fallback

During assessment of the input, you might have determined the following:

- Different portions of the content are in different color spaces. All those color spaces are supported.

- The metadata for one color space is either inaccurate everywhere, or is sometimes accurate and sometimes inaccurate.
- The metadata for content for all the other color spaces is accurate.

For example, the input has Rec. 601 content that has portions that are inaccurately marked. It also has portions that are missing, unknown, or unsupported. The input also has HDR10 content and HLG content that is accurately marked.

You have this option for handling the metadata in the output:

Correct the metadata

Follow the procedure in the section called "Step 3: Set up inputs", and set the key fields as follows:

- **Color space** field – Set to the color space that has inconsistent metadata (Rec. 601 in the above example).
- **Color space usage** field – Set to **FALLBACK**

During ingest, MediaLive will create metadata of the specific color space for all missing, unmarked, and unknown video content. It won't change any supported color space metadata. (It will *fall back* to the existing metadata.) Therefore, it won't change the accurately marked Rec. 601 or the accurately marked HDR10 or HLG content.

After ingest, all the content in the input will be consistently marked, even though the content is in several color spaces.

Scenario D – Metadata can't be corrected

During assessment of the input, you might have determined the following:

- Different portions of the content are in different color spaces. All those color spaces are supported.

- The metadata is inaccurate for more than one color space. (Compare this to scenario C, where the metadata is inaccurate only for one color space.)

Or you might have determined the following:

- The content provider can't provide accurate information about the color space or its metadata.

You have this option for handling the metadata in the output:

Remove the metadata

There is no way to clean up this content because MediaLive can correct the metadata for only one color space. In this scenario, the metadata is inaccurate in different types of color space.

You can't force the color space metadata. For example, you can't force it to Rec. 601, because sometimes will correctly identify the accompanying color space, but sometimes it won't. Inaccurate metadata will result in an inaccurate conversion (if you convert the color space in the output), or in an inferior viewing experience (if you pass through the color space in the output).

Follow the procedure in the section called "Step 3: Set up inputs", and set the key fields as follows:

- **Color space** field – Set to **FOLLOW**
- **Color space usage** field – MediaLive ignores this field.

During processing, MediaLive won't read the metadata.

You won't be able to convert the color space in any outputs, even for other inputs that have correct color space metadata.

Step 3: Set up inputs to correct metadata

In the previous step, you identified how to correct color space metadata in each input. This section describes how to set up each input for the required correction.

> **Note**
>
> This section assumes that you are familiar with creating or editing a channel, as described in *Setup: Creating a channel* .

To set up each input attached to the channel

1. On the **Create Channel** page, in the **Input attachments** section, for **Video selector**, choose **Video selector**.
2. Set the appropriate values for **Color space** and **Color space usage**. See the table after this procedure.
3. This step applies only if you chose **HDR10** and the attached input is for a MediaLive device such as AWS Elemental Link, and you plan to convert the content to another color space. You must specify the values for the Max CLL and Max FALL for the content. You should have obtained this information from the content provider.

In the **Max CLL** field and the **Max FALL** field, enter the values.

In the following table, each row shows a valid combination of the two fields and the result of that combination.

Color space field	Color space usage field	Result
FOLLOW	This field is ignored.	Passthrough. MediaLive doesn't change the color space metadata.
REC_601 or **REC_709** or **HDR10** or **HLG** or **Dolby Vision 8.1**	**Force**	Cleanup. MediaLive marks all the content as using the specified color space.
REC_601 or **REC_709** or **HDR10** or **HLG** or **Dolby Vision 8.1**	**Fallback**	Cleanup. MediaLive marks the content as using the specified color space only for portions of the content that are unmarked or marked as unknown or marked with an unsupported color space.

Configuring color space handling in each output

In the section called "Options for handling", you should have identified how you want to handle the color space in each output. When you assessed the inputs, you might have adjusted your plan. You should now have a clear plan for handling the color space in the outputs.

Topics

Step 1: Decide about enhanced VQ mode

You must decide if you should enable enhanced VQ mode in the output. This mode applies only to outputs that use H.264.

In the following table, find the planned handling in the first column, then read across to identify the action to take. To enable enhanced VQ mode, see the section called “Video – enhanced VQ”.

Planned conversion	Details	Action
Convert to SDR	The inputs contain both SDR and HDR content.	You must enable the mode.
Convert to SDR	The inputs contain only SDR content. For example, all the inputs are Rec. 709, and you want to convert the content to Rec. 601.	You don't need to enable the mode.
Any handling	There is no HDR10 or HLG in any of the inputs.	You don't need to enable the mode.
Any handling	You have already enabled enhanced VQ to improve the video quality.	Leave the mode enabled.

Step 2: Set up outputs to process color space

Follow this procedure to configure color space handling in each output. You can set up each output with different color space handling. For example, you can create one output that passes through the original color space, and another that converts it.

> **Note**
>
> This section assumes that you are familiar with creating or editing a channel, as described in *Setup: Creating a channel* .

Topics

Setting up for passthrough

You can set up to pass through the source color space in one or more outputs. The key fields to set are **Color space** and **Color metadata**.

1. On the **Create channel** page, in the **Output groups** section, choose the output that contains the video.
2. Display the **Stream settings** section, and choose the **Video** section.
3. For **Codec settings**, choose a codec. For information about the color spaces that each codec supports, see the section called “Supported output codecs”.
4. Choose **Codec details**. More fields appear. Choose **Additional settings**. More fields appear.

   In **Color metadata**, choose **Insert** or **Ignore** to specify how you want to handle the color space metadata.
5. Choose **Color space**. The **Color space settings** field appears. Choose **Color space passthrough**. (Or choose **Don't include**, which is equivalent to **Color space passthrough**.)

Setting up to convert

You can set up to convert the color space in one or more outputs. There are several fields that must each be set in a specific way.

You can set up to convert the source color space in one or more outputs.

1. On the **Create channel** page, in the **Output groups** section, select the output that contains the video.

2. Display the **Stream settings** section, and choose the **Video** section.
3. Complete the **Width** and **Height** fields to specify a valid resolution. Make a note of whether you are specifying an SD, an HD, or a UHD resolution.
4. In **Codec settings**, choose a codec. For information about the color spaces that each codec supports, see the section called “Supported output codecs”.
5. Choose **Codec details**. More fields appear. Set the **Profile**, **Tier** (for H.265 only), and **Level** fields:
   - If the resolution is SD, enter values that suit your requirements.
   - If the resolution is an HD or UHD resolution, set the tier and level to suit your requirements, and set the profile as follows:
     - If the output color space will be an HDR color space, you must choose one of the profiles that has **10BIT** in the name.
     - If the output color space will be an SDR color space, you can choose any profile.
6. Choose **Color space**. The **Color space settings** field appears.

   Set the field to the color space to convert to.

   If you choose **HDR10**, the **Max CLL** and **Max FALL** fields appear. Complete these fields to set the display metadata.
7. Go back to **Codec details** and choose **Additional settings**. More fields appear, including **Color metadata**. In **Color metadata**, choose **Insert** or **Ignore** to specify how you want to handle the color space metadata.

Results for different color space handling

This section describes how MediaLive handles the color space and color space metadata that it encounters in the source input, depending on how you set up the color space in the output.

Topics

Result when passing through color space

Read this section if you set up one or more outputs to pass through the color space. The following table shows how MediaLive handles each type of color space that it encounters in the source.

Color space that MediaLive encounters	How MediaLive handles the color space
Content in any color space that MediaLive supports	Doesn't touch the color space or brightness (the pixel values) in the output. Passes through any of the three sets of metadata that are present.
Content in a color space that MediaLive supports, but that isn't supported for the output codec.	This conversion isn't supported. After conversion, the color map of the content will be completely wrong.
Content marked with unknown or an unsupported color space	Doesn't touch the color space or brightness (the pixel values) in the output. Leaves the content as marked with the unknown color space. Passes through any brightness metadata and display metadata.
Content with no color space metadata	Doesn't touch the color space or brightness (the pixel values) in the output. Leaves the content as unmarked (no color space metadata).

Result when converting color space to SDR

Read this section if you set up one or more outputs to convert the color space to Rec. 601 or Rec. 709. The following table shows how MediaLive handles each type of color space that it encounters in the source.

Color space that MediaLive encounters	How MediaLive handles the color space
Content in the same SDR color space	• Doesn't touch the color space (the pixel values) in the output. • Passes through the color space metadata. • Passes through the brightness metadata.
Content in the other SDR color space	• Converts the content to the chosen SDR color space and brightness function. The conversion maps the pixels to code values that represent the same color as the original code values. • Changes the color space metadata to specify the new color space. • Passes through the brightness metadata. This is appropriate because the two SDR color spaces use the same brightness function.
Content in HDR10	When the output codec is H.264 and if you have enabled enhanced VQ, MediaLive does the following: • Converts the content to the chosen SDR color space and brightness function. The conversion fits the colors into the smaller color space. • Changes the color space metadata to specify the new color space. • Changes any brightness metadata to specify the new standard. • Removes any display metadata. After the conversion, the content complies completely with the new color space. The

Color space that MediaLive encounters	How MediaLive handles the color space
	color will be less rich. The color will match the new brightness function. If you haven't enabled VQ, MediaLive doesn't convert anything. It passes through the color space metadata, any brightness metadata, and any display metadata.
	When the output codec is H.265, MediaLive does the following: • Converts the content to the chosen SDR color space and brightness function. The conversion fits the colors into the smaller color space. • Changes the color space metadata to specify the new color space. • Changes any brightness metadata to specify the new standard. • Removes any display metadata. After the conversion, the content complies completely with the new color space. The color will be less rich. The color will match the new brightness function.

Color space that MediaLive encounters	How MediaLive handles the color space
Content in HLG	When the output codec is H.264 and if you have enabled enhanced VQ, MediaLive does the following: • Converts the content to the chosen SDR color space and brightness function. The conversion fits the colors into the smaller color space. • Changes the color space metadata to specify the new color space. • Changes any brightness metadata to specify the new standard. After the conversion, the content complies completely with the new color space. The color will be less rich. The color will match the new brightness function. If you haven't enabled VQ, MediaLive doesn't convert anything. It passes through the color space metadata, and any brightness metadata.

Color space that MediaLive encounters	How MediaLive handles the color space
	When the output codec is H.265, MediaLive does the following: • Converts the content to the chosen SDR color space and brightness function. The conversion fits the colors into the smaller color space. • Changes the color space metadata to specify the new color space. • Changes any brightness metadata to specify the new standard. After the conversion, the content complies completely with the new color space. The color will be less rich. The color will match the new brightness function.
Content in Dolby Vision 8.1	This conversion isn't supported. After conversion, the color map of the content will be completely wrong.

Color space that MediaLive encounters	How MediaLive handles the color space
Content marked with an unknown or unsupported color space	We can't make any promises about how MediaLive will handle input that is in an unsupported color space. Any of the following might apply: • MediaLive might be able to ingest the input, and to pass through the color space and all the color space metadata. • Or it might ingest the input but produce unacceptable output. • Or it might fail to ingest the input, so that the event follows the input loss behavior routine (for example, it might display a slate in the output).
Content with no color space metadata	• Doesn't touch the color space (the pixel values) in the output. • Leaves the content as unmarked (no color space metadata). • Passes through any brightness metadata and display metadata.

Result when converting color space to HDR10

Read this section if you set up one or more outputs to convert the color space to HDR10. The following table shows how MediaLive handles each type of color space that it encounters in the source.

Color space that MediaLive encounters	How MediaLive handles the color space
Content in an SDR color space	• Converts the content to the new color space and brightness function. The conversio n maps the pixels to code values that

Color space that MediaLive encounters	How MediaLive handles the color space
	represent the same color as the original code values. • Changes the color space metadata to specify the new color space. • Changes the brightness metadata to specify the new standard. • If you completed **Max CLL** and **Max FALL**, the values in those fields are inserted in the display metadata. This conversion fits the smaller SDR color space into the larger HDR color space and maps the pixels to new code values that represent the same color. The conversion doesn't actually make the existing color richer. However, the bright parts of the content are brighter, and the dark parts are darker.
Content in HDR10	• Doesn't touch the color space (the pixel values) in the output. • Passes through the color space metadata. • Passes through the brightness metadata and the display metadata.

Color space that MediaLive encounters	How MediaLive handles the color space
Content in HLG	• There is no color space conversion—there is no change to the pixel values of the content. This is appropriate because HDR10 and HLG use the same color space (they are different only in brightness function and display metadata). • Converts the content to the new brightness function. • Changes the color space metadata to the new color space. • Change the brightness metadata to specify the new standard. • If you completed **Max CLL** and **Max FALL**, MediaLive inserts the values from those fields into the display metadata.
Content in Dolby Vision 8.1	This conversion isn't supported. After conversion, the color map of the content could be completely wrong.
Content marked with an unknown or unsupported color space	We can't make any promises about how MediaLive will handle source content that is in an unsupported color space. Any of the following might apply: • MediaLive might be able to ingest the input, and to pass through the color space and all the color space metadata. • Or it might ingest the input but produce unacceptable output. • Or it might fail to ingest the input, so that the event follows the input loss behavior routine (for example, it might display a slate in the output).

Color space that MediaLive encounters	How MediaLive handles the color space
Content with no color space metadata	• No change to the pixel values of the content. • Leaves the content as unmarked. • Passes through any brightness metadata and display metadata.

Result when converting color space to Dolby Vision 8.1

Read this section if you set up one or more outputs to convert the color space to Dolby Vision 8.1. The following table shows how MediaLive handles each type of color space that it encounters in the source.

Color space that MediaLive encounters	How MediaLive handles the color space
Content in HDR10	When you convert suitable content to Dolby Vision 8.1, MediaLive makes the following changes: • It doesn't change the pixel values, because HDR10 and Dolby Vision 8.1 both use the same color space. • It changes the color space metadata to identify the new color space. • It applies the new brightness function to the content. • It calculates the Dolby Vision 8.1 display metadata for the content. After the conversion, the color space hasn't changed. However, the bright parts of the content are brighter, and the dark parts are darker.

Color space that MediaLive encounters	How MediaLive handles the color space
Content in any other supported color space	MediaLive passes through the color space and color space metadata for that portion,
Content marked with an unknown or unsupported color space	Converting non-HDR10 content to Dolby Vision 8.1 doesn't comply with the usage intended by Dolby Vision 8.1. After conversio n of the color space, the color map of the content will be completely wrong.
Content with no color space metadata	

Result when removing color space metadata

Read this section if you set up one or more outputs to pass through the color space or convert the color space and you chose to remove the color space metadata. The following table shows how MediaLive handles each type of color space that it encounters in the source.

Color space that MediaLive encounters	How MediaLive handles the color space
Content in any color space that MediaLive supports Content with no color space metadata	• Doesn't touch the color space or brightness (the pixel values) in the output. • Removes the color space metadata. The output won't contain any color space metadata, brightness metadata, or display metadata.
Content marked with an unknown or unsupported color space	We can't make any promises about how MediaLive will handle input that is in an unsupported color space. Any of the following might apply: • MediaLive might be able to ingest the input, and to pass through the color space and all the color space metadata.

Color space that MediaLive encounters	How MediaLive handles the color space
	• Or it might ingest the input but produce unacceptable output. • Or it might fail to ingest the input, so that the event follows the input loss behavior routine (for example, it might display a slate in the output).

Reference: Location of fields

Read this section if you know how to handle color space in MediaLive, and you only need a reminder of where the fields are located in the MediaLive Console.

Topic	Location on the Channel page		Field
Input handling	**Input attachments**	**Video Selector**	**Color space**
			Color space usage
Enter the display metadata for an input from a AWS Elemental Link device	**Input attachments**	**Video Selector**, then **Color space settings**	**Max CLL**
			Max Fall
Output, configure the video codec	**Output groups**, then **Outputs**	**Stream settings**, then **Video**	**Codec settings**
		Stream settings, then **Video**, then **Codec settings**, then **Codec details**	**Profile**
			Tier
			Level
Output, convert the color space	**Output groups**, then **Outputs**	**Stream settings**, then **Video**, then **Color space**	**Color space settings**

Topic	Location on the Channel page		Field
Output, include or omit color space metadata	**Output groups**, then **Outputs**	**Stream settings**, then **Video**, then **Codec settings**, then **Codec details**, then **Additional settings**	**Color metadata**
Output, specify display metadata to include, only if you are converting to HDR10	**Output groups**, then **Outputs**	**Stream settings**, then **Video**, then **Color space**, then **Color space settings**	**Max CLL** **Max Fall**
Output, set up enhanced VQ, only if the output codec is H.264	**Output groups**, then **Outputs**	**Stream settings**, then **Video**, then **Codec settings**, then **Additional encoding settings**	**Quality level** **Filter settings**

Setting up enhanced VQ mode

Enhanced VQ is an optional mode that affects the video quality of outputs. It affects the video encode where both of the following apply:

- The encode uses H.264 (AVC).
- The encode uses QVBR or CBR rate control mode.

Enhanced VQ applies as follows:

- It doesn't apply to Frame Capture output groups.
- It does apply to Multiplex output groups. For this type of output group, you *must* enable the mode.
- It does apply to the other types of output groups. For these types, you can *optionally* enable the mode.

- It is required if you want to convert an HDR color space to an SDR color space in an output that uses H.264. Even if the encode uses VBR, you must enable the mode to obtain this color space conversion. For more information, see the section called "Configure enhanced VQ".

> **Note**
>
> If the rate control mode is VBR, there is no benefit to setting up enhanced VQ mode. But the channel would still incur the costs for enhanced VQ.

For more information about the benefits of enhanced VQ mode, see Benefits of enhanced VQ.

For information on charges for using this mode, see the MediaLive price list.

> **Note**
>
> This section assumes that you are familiar with creating or editing a channel, as described in *Setup: Creating a channel* .
> The fields in the console for setting this mode are in the **Codec settings** section of the video **Stream settings** in each output. To review the step where you complete these fields, see the section called "Set up video".

To enable enhanced VQ

You can enable enhanced VQ in any video encode that uses H.264 (AVC) as the codec.

1. In the **Output groups** section of the **Create channel** page of the MediaLive console, in the **Stream settings** pane, choose **Video**.
2. In the **Codec settings** section, expand the **Additional encoding settings** section.
3. For **Quality level**, choose **ENHANCED_QUALITY**.
4. (Optional) For **Filter settings**, choose **Temporal filter**. Or to omit the filter, choose **Don't include**.

   For information about the benefits of filters, see Benefits of the temporal filter.
5. If you choose **Temporal**, optionally change the default strength, and optionally enable sharpening. For details about a field on the MediaLive console, choose the **Info** link next to the field.

Benefits of enhanced VQ

When enhanced VQ is enabled, MediaLive can produce slightly better video quality without an increase in the bitrate (the bitrate fields in the **Rate control** section under **Codec settings**).

You can therefore use enhanced VQ in one of two ways:

- You can choose to take advantage of the improved video quality. Typically, the main improvement is to smooth out complex transitions in high-motion video content.
- You can choose to lower the bitrate (by perhaps 5%) and maintain the original target video quality. Doing so lowers the bandwidth requirements for the output.
  - To change the bitrate when the rate control mode is QVBR, change the **Max bitrate**.
  - To change the bitrate when the rate control mode is CBR, change the **Bitrate**.

Benefits of the temporal filter

The temporal filter is useful for both source content that is noisy (when it has excessive digital artifacts) and source content that is clean.

When the content is noisy, the filter cleans up the source content before the encoding phase, with these two effects:

- It improves the output video quality because the content has been cleaned up.
- It decreases the bandwidth because MediaLive doesn't waste bits on encoding noise.

When the content is reasonably clean, the filter tends to decrease the bitrate, especially when the rate control mode is QVBR.

Setting the rate control mode

This feature does not apply to the video in a Frame Capture output.

You can configure the rate control mode when you set up the video as part of creating a channel. This feature lets you control the quality and bitrate of the video.

> **Note**
>
> The information in this section assumes that you are familiar with the general steps for creating a channel, as described in *Setup: Creating a channel* , and specifically with setting up the video.

When encoding visually complex video (such as high-motion sports events with brightly dressed crowds in the background), there is always a trade-off between high video quality and low bitrate. Higher video quality requires higher bitrate. There is less trade-off with visually simple video such as cartoons.

AWS Elemental MediaLive offers several options that provide different balances of video quality versus bitrate.

To set the rate control mode and bitrate for the output

1. On the **Stream settings** pane, for **Video**, for **Codec settings**, choose **H264**.
2. In the **Rate Control** section, for **Rate control mode**, choose **QVBR** or **CBR** or **VBR**. The default mode is CBR. For information about choosing the best option and about completing the other fields in the **Rate control mode** section, see the sections below.

Topics

Quality-defined variable bitrate mode (QVBR)

With quality-defined variable bitrate mode (QVBR), MediaLive aims for a specific quality and uses only the bitrate that it needs to reach that quality. Video quality will match the specified quality except when the video is very complex. In this case, when it's not possible to reach the desired quality without exceeding the maximum bitrate, MediaLive observes the maximum bitrate. This means that the video doesn't achieve the desired quality.

We recommend this mode if you or your viewers pay for bandwidth, for example, if you are delivering to a CDN such as Amazon CloudFront or if your viewing users are on mobile networks.

With QVBR mode, you can specify a target quality, or you can let MediaLive determine the target quality.

Option 1: Setting a target quality

To set up in QVBR mode with a target quality that you specify, complete the fields as follows:

- **Max bitrate**. See the table that follows this list.
- **Quality level**. See the table that follows this list.
- You must also set the Bitrate field. Enter the same value as you enter in **Max bitrate**.

  This field has no effect on quality level in QVBR mode, but MediaLive does use it for calculating the output charges for this output. For more information about charges, see the MediaLive price list. If you leave **Bitrate** empty, MediaLive calculates charges using the value in the **Max input** bitrate in the **Input specifications** section of the channel configuration.
- Set **Buffer size** to twice the maximum bitrate.
- Set **Buffer fill percentage** to 90%.
- Ignore the other fields in this section. They aren't used for QVBR.

Values to use: Set the **Max bitrate** and **Quality level** for your most important viewing devices. See the following table for suggestions.

Viewing Device	Quality Level	Max Bitrate
Primary Screen	8 to 10	4,000,000 to 6,000,000
PC or Tablet	7	1,500,000 to 3,000,000
Smartphone	6	1,000,000 to 1,500,000

How it works: The bitrate can change with each frame (in order to obtain at least the specified quality), but it can't exceed the maximum bitrate. The encoder does not attempt to maintain an average bitrate. It always reaches the maximum bitrate if that is necessary to obtain the specified quality. On the other hand, if the quality can be obtained with lower bitrates, the encoder doesn't use a higher bitrate.

Option 2: Letting MediaLive determine the quality level

To set up in QVBR mode with a target quality that MediaLive determines, complete the fields as follows:

- Leave the **QVBR quality level** field empty.
- In **Max bitrate**, enter the maximum rate you want the output to use.
- You must also set the Bitrate field. Enter the same value as you enter in **Max bitrate**.

  This field has no effect on quality level in QVBR mode, but MediaLive does use it for calculating the output charges for this output. For more information about charges, see the MediaLive price list. If you leave **Bitrate** empty, MediaLive calculates charges using the value in the **Max input** bitrate in the **Input specifications** section of the channel configuration.
- Set **Buffer size** to twice the maximum bitrate.
- Set **Buffer fill percentage** to 90%.

How it works: You don't specify a target quality. Instead, MediaLive infers the quality you want based on the following fields you completed:

- The output video resolution (the values in the **Height** and **Width** fields that are also in this Video section).
- The maximum bitrate.

The bitrate can change with each frame (in order to obtain at least the quality that MediaLive has identified), but it can't exceed the maximum bitrate. The encoder does not attempt to maintain an average bitrate. It always reaches the maximum bitrate if that is necessary to obtain the identified quality. On the other hand, if the quality can be obtained with lower bitrates, the encoder doesn't use a higher bitrate.

Variable bitrate mode (VBR)

With variable bitrate mode (VBR), you specify an average bitrate and a maximum bitrate. Video quality and bitrate vary, depending on the video complexity.

Choose VBR instead of QVBR if you want to maintain a specific average bitrate over the duration of the channel. If bitrate does not need to be constrained, then consider using QVBR.

To set up VBR mode, complete the fields as follows:

- **Bitrate** (average bitrate). Try to assess the expected complexity of the video, and set a suitable average bitrate.

  If you leave **Bitrate** empty, MediaLive sets the average bitrate to 5 Mbps.

  The value you enter in **Bitrate** also affects the output charges for this output. If you leave **Bitrate** empty, MediaLive calculates charges using the value in the **Max input** bitrate in the **Input specifications** section of the channel configuration. For more information about charges, see the MediaLive price list.
- Set the **Max bitrate** to accommodate expected spikes.
- Set **Buffer size** to twice the maximum bitrate.
- Set **Buffer fill percentage** to 90%.
- Ignore the other fields in this section. They aren't used for VBR.

How it works: The bitrate can change with each frame (in order to obtain the best quality) but it can't exceed the specified maximum bitrate. The encoder also ensures that as the channel progresses, the stream meets the specified average bitrate. This mode is useful when you expect short spikes in the complexity of the video. The encoder aims for the average bitrate but spikes to the maximum bitrate for a short time when necessary.

Constant bitrate mode (CBR)

With constant bitrate mode (CBR), you specify a bitrate. Video quality varies, depending on the video complexity.

Choose CBR only if you distribute your assets to devices that cannot handle variable bitrates.

But if it's acceptable for the bitrate to occasionally differ from a specified rate, then consider using VBR or QVBR. Over the duration of the channel, you might obtain both a lower bitrate and better quality with VBR or QVBR.

To set up CBR mode, complete the fields as follows:

- **Bitrate**. Set the **Bitrate** to balance the video quality and the output bitrate. If you leave this field empty, MediaLive sets the bitrate to 5 Mbps.

  The value you enter in **Bitrate** also affects the output charges for this output. If you leave **Bitrate** empty, MediaLive calculates charges using the value in the **Max input** bitrate in the **Input**

specifications section of the channel configuration. For more information about charges, see the MediaLive price list.

- Set **Buffer size** to twice the bitrate.
- Set **Buffer fill percentage** to 90%.
- Ignore the other fields in this section. They aren't used for CBR.

How it works: The output always matches the specified bitrate. Sometimes that bitrate results in higher-quality video, and sometimes it results in lower-quality video.

Delivering outputs via your VPC

You can set up a channel to have output endpoints in Amazon Virtual Private Cloud (Amazon VPC). This delivery mode is useful if an important output destination for your channel is an address in your VPC.

The output destination in your VPC is typically an address in Amazon EC2. It could also be a bucket in Amazon Simple Storage Service (Amazon S3), if you have set up VPC endpoints for Amazon S3. You might want to send output to your VPC so that you can perform post-processing, or so that you can deliver the video over AWS Direct Connect.

If you don't have a VPC, you can stop reading this section. You will always set up the channel in the regular way, with endpoints in MediaLive. You don't have to perform any special setup in order to set up channels in the regular way.

Rules and constraints

The following rules apply to a channel that is set up for delivery via your VPC:

- You can't change an existing channel to either start delivering to your VPC or stop delivering via your VPC.
- The channel class can be either standard or single-pipeline.
- You can't change the channel class on an existing channel.
- You can't include multiplex output groups in the channel.
- The channel can have output groups with destinations in your VPC, with destinations at other locations (such as AWS Elemental MediaPackage), and with destinations on the public internet.

> **Note**
>
> The information in this section assumes that you are very familiar with Amazon Virtual Private Cloud, with AWS PrivateLink, with AWS Direct Connect, and with general networking practices.

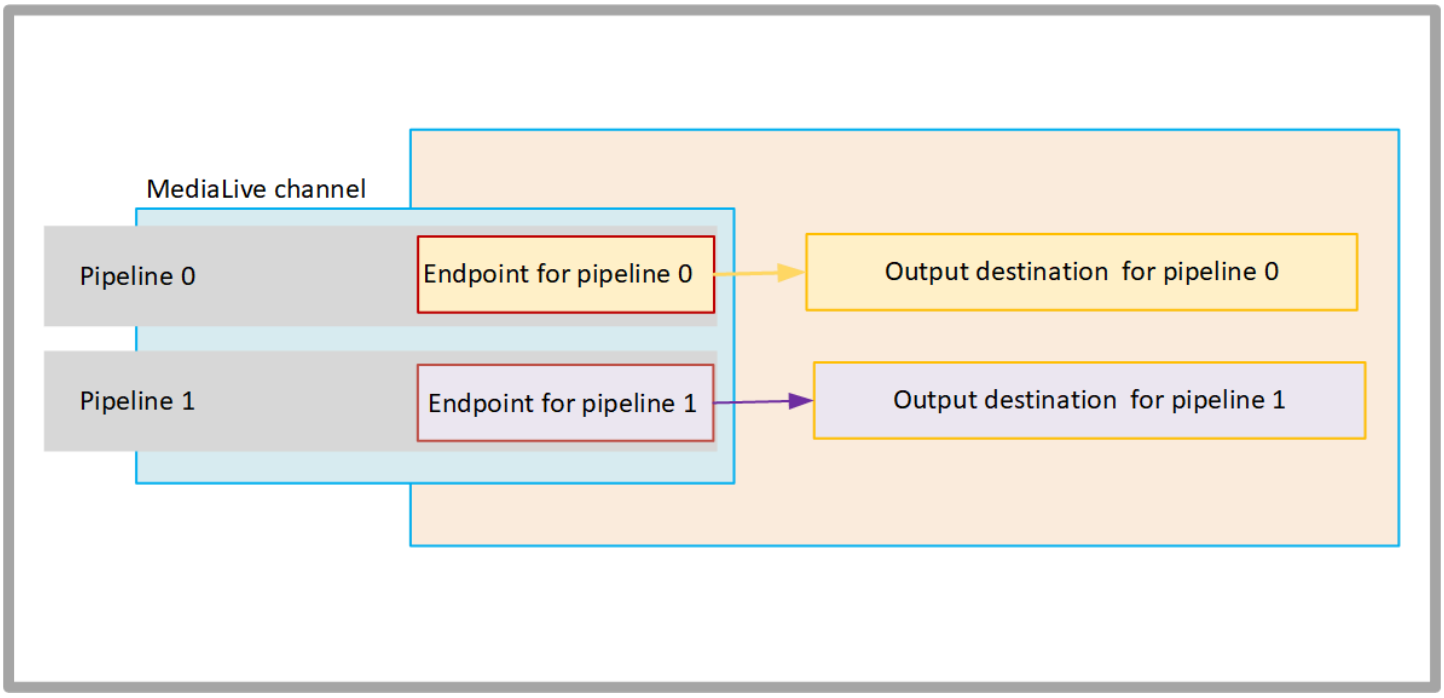
Topics

How VPC delivery works

VPC delivery applies to each MediaLive channel. You can have some channels that deliver via your VPC, and other channels that deliver in the regular way.

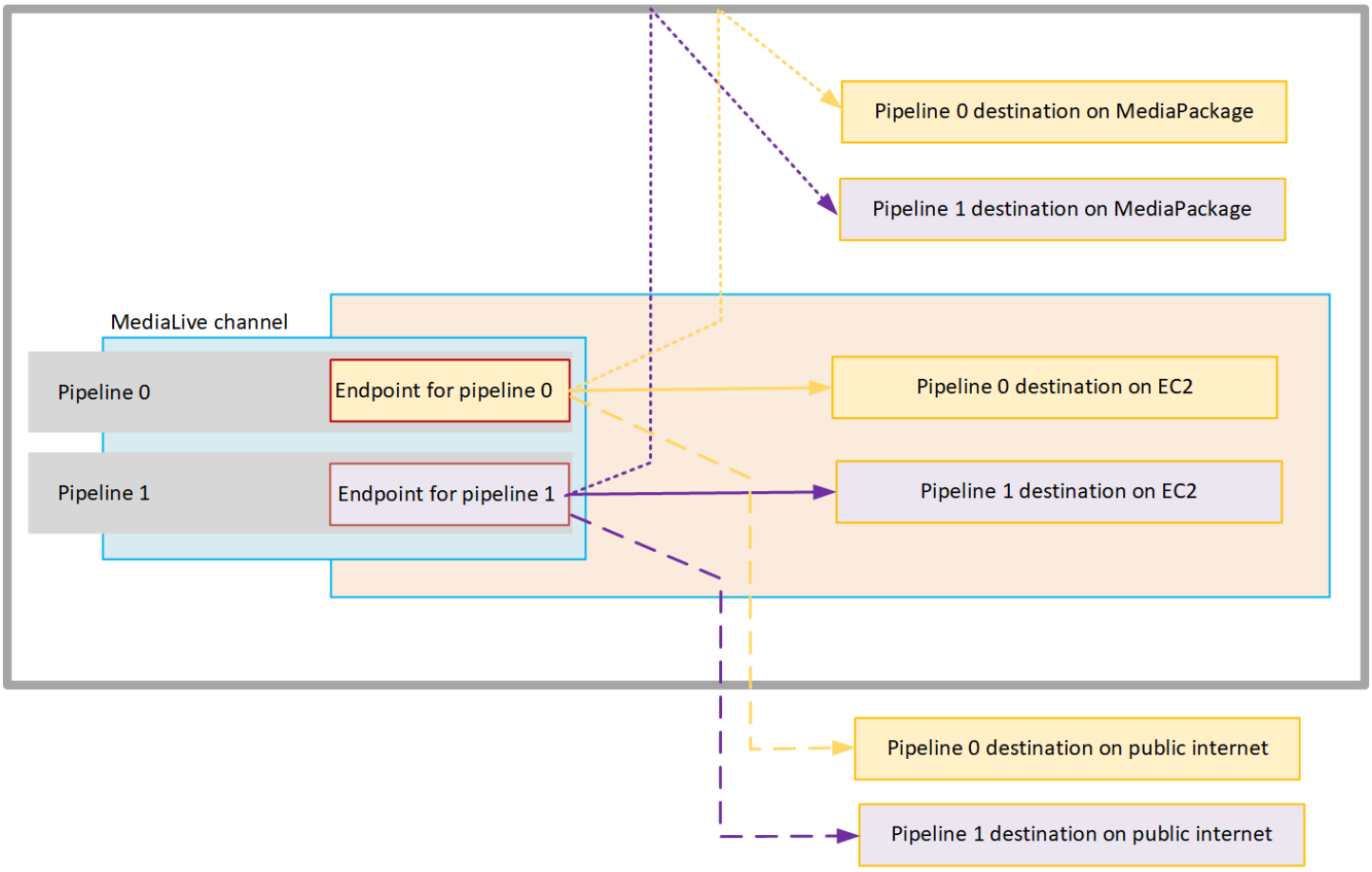
With VPC delivery, the endpoints for the channel are in your VPC, rather than in the VPC that MediaLive owns. This setup provides benefits including improved security, because the output doesn't have to go to the boundary of the public internet to reach the output destinations that are in your VPC.

The following diagram illustrates how VPC delivery works. The blue box is a channel with two pipelines. The orange box is your VPC. Notice that the endpoints for the two pipelines are in your VPC. In this example, you have only one output group, with a destination in EC2 in your VPC. This output group might be an HLS output group being sent to an HTTP server on your EC2 instance.



The following diagram illustrates a channel with three output groups:

- The destination for one output group is on your EC2 instance.
- The destination for the output shown at the top is on MediaPackage. The output leaves the pipeline endpoint, goes to the boundary of AWS (the gray box), and comes back in, to the destination on AWS Elemental MediaPackage.
- The destination for the output shown at the bottom is on the public internet. The output leaves the pipeline, then leaves AWS and enters the public internet.



You set up for delivery to your VPC as follows:

- Identify subnets and security groups in your VPC for the channel endpoints.
- Identify subnets and security groups for the output destinations, for those outputs groups with destinations in your VPC.
- Determine if you need to identify Elastic IP addresses to associate with the channel.
- Check the permissions that are required for your trusted entity role for MediaLive. You must update the role if your channel uses a custom trusted entity role, rather than the built-in MediaLiveAccessRole role that is available through the console. For more information, see the section called “Reference: summary of trusted entity access”.
- Update the IAM policies for users. For more information, see the section called “Reference: summary of user access”.
- When you create a channel, you must include this subnet, security group, and Elastic IP address information in the channel configuration.

The following sections describe this setup in detail.

Getting ready

An Amazon VPC user must set up the VPC and identify subnets and security groups for the channel.

To set up the VPC

1. Provide your Amazon VPC user with the following guidelines:

   - Guideline for the subnets and Availability Zones – See the section called "Identifying subnet and Availability Zone requirements"
   - Guideline for the security group for channel endpoints subnets – The security group or groups must follow these rules:
     - The combined rules of the security groups must allow outbound traffic from the endpoint to all the output destinations. These destinations might be on your VPC, destinations on AWS services, and destinations on the public internet.
   - Guideline for the security group for destination subnets – The security group or groups must follow these rules:
     - The combined rules of the security groups must allow inbound traffic from the channel endpoints.
2. Determine if you need to identify EIPs to associate with the channel. If the channel has output groups with destinations outside your VPC, you must provide a mechanism for the content to leave the VPC. One way to do this is to associate EIPs with the channel endpoints. These endpoints appear in the diagram in the section called "How VPC delivery works" Speak to the Amazon VPC user about your requirements.

   If you decide to associate EIPs with the channel endpoints, identify those EIPs.
3. After the Amazon VPC user has performed the setup, obtain the following information:

   - The ID of the VPC or VPCs.
   - The IDs of the subnets and Availability Zones for the channel endpoints.
   - The IDs of the subnets and Availability Zones for the destinations.
   - The IDs of the security groups for the subnets.

- The elastic IP address to associate with the elastic network interfaces of the channel endpoints.

4. Delivery via the VPC depends on appropriate setup for routing and DNS. of the VPC network. Provide the Amazon VPC user with these guidelines:

   - If you expect addresses with a domain name to reach the VPC, or if you expect the VPC to reach addresses with a domain name, you must set up a DNS to resolve those domain names. This requirement applies equally to AWS services that might have domain names.
   - If any communication with the public internet is expected, you will need either a NAT or an Internet Gateway in your VPC.
   - Inside the VPC, you must configure routing tables, to allow communication between the subnets you intend to use.
   - All IP addresses must be IPV4.

Setting up for VPC delivery

> **Note**
>
> The information in this section assumes that you are familiar with the general steps for creating a channel. It also assumes that you have read *Setup: Preparing upstream and downstream* and have planned the workflow for your channel.

To set up for VPC delivery

Follow these steps at some point when you are creating the channel.

1. On the **Create channel** page, choose **Channel and input details** in the navigation pane.
2. Complete the **Output delivery** section:

   - **Delivery method** – Choose **VPC**.
   - **VPC settings** – Choose **Select subnets and security groups**.
   - **Subnets** – Choose one of the subnets that you obtained. The dropdown list shows subnets in all VPCs, identified as follows:

     ```
     <subnet ID> <Availability Zone of subnet> <IPv4 CIDR block of subnet>
     <VPC ID> <Subnet tag called "Name", if it exists>
     ```

For example:

subnet-1122aabb us-west-2a 10.1.128.0/24 vpc-3f139646 Subnet for VPC endpoints

If the list of subnets is empty, choose **Specify custom VPC**, and enter the subnet ID in the field. (You need to enter only the subnet ID, for example, **`subnet-1122aabb`**.)

MediaLive associates this subnet with pipeline 0.

- If your channel is a standard channel, add another subnet. Still in **Subnets**, choose the second subnet. This second time, the dropdown list shows only the subnets in the same VPC as the first subnet.

  MediaLive associates this subnet with pipeline 1.

- **Security groups** – Choose the security group or groups that you obtained, following the same process as for the subnets. The dropdown list shows security groups belonging to the VPC that you chose, identified as follows:

  `<`*`security group ID`*`> <`*`description attached to this security group`*`> <`*`VPC ID`*`>`

- **EIPs for endpoints** – If applicable, enter the Elastic IP addresses that you obtained. MediaLive takes the first Elastic IP address that you specify and associates it with pipeline 0. It associates the second Elastic IP address (if applicable) with pipeline 1.

3. Follow these guidelines when you create the output groups in the channel:

   - For the channel output groups that have destinations in your VPC or on Amazon S3, obtain the URL or bucket path. You don't have to modify the destination syntax. If the Amazon VPC user has set up the routing correctly, the outputs will successfully find these outputs in the VPC.
   - For the channel output groups that have destinations that are not in your VPC, follow the usual procedure. You don't have to modify the destination syntax. If the Amazon VPC user has set up the routing correctly, the outputs will successfully find the outputs that are outside the VPC.

Result

When you set up for delivery via your VPC, MediaLive creates one or two elastic network interfaces in your VPC. It creates one elastic network interface for a single-pipeline channel, and two for a standard channel.

If you choose to use Elastic IP addresses, MediaLive also associates those Elastic IP addresses with the elastic network interface.

You can view the setup of the delivery point in the details for the channel.

Changing the setup

If you have set up a channel for VPC delivery, note the following:

- You can't change an existing channel to either start delivering via your VPC or stop delivering via your VPC.
- You can't change the channel class on an existing channel that is set up for delivery via your VPC.
- If you add another input that uses your VPC, make sure that it follows the already established rules for VPCs, subnets, and Availability Zones.
- If you delete the channel or if you delete all the output groups, MediaLive deletes the elastic interface points that it created in your Amazon EC2 instance.

Identifying subnet and Availability Zone requirements

Subnets and Availability Zones apply as follows:

- **Inputs** – Some MediaLive input types are in your VPC, which means that they are in a specific subnet. For example, an RTMP input can be in your VPC. For more information, see the section called “Input types, protocols, upstream systems”.
- **Endpoints** – The channel endpoints are in a subnet.
- **Destinations** –The IP addresses for outputs in the VPC are in a subnet. You identify IP addresses (and their implied subnets) when you plan the downstream system.

You must identify the VPCs and subnets for the MediaLive endpoints and for those of your output destinations that are an address in your VPC. You must consider the following:

- You must make sure that the setup follows the rules for allocation across subnets and across Availability Zones. See the section called "Use case A – no VPC inputs" and the section that follows it.
- Each subnet must have a private CIDR block (a range of IP addresses).
- Each subnet must have at least two unused addresses in that block.

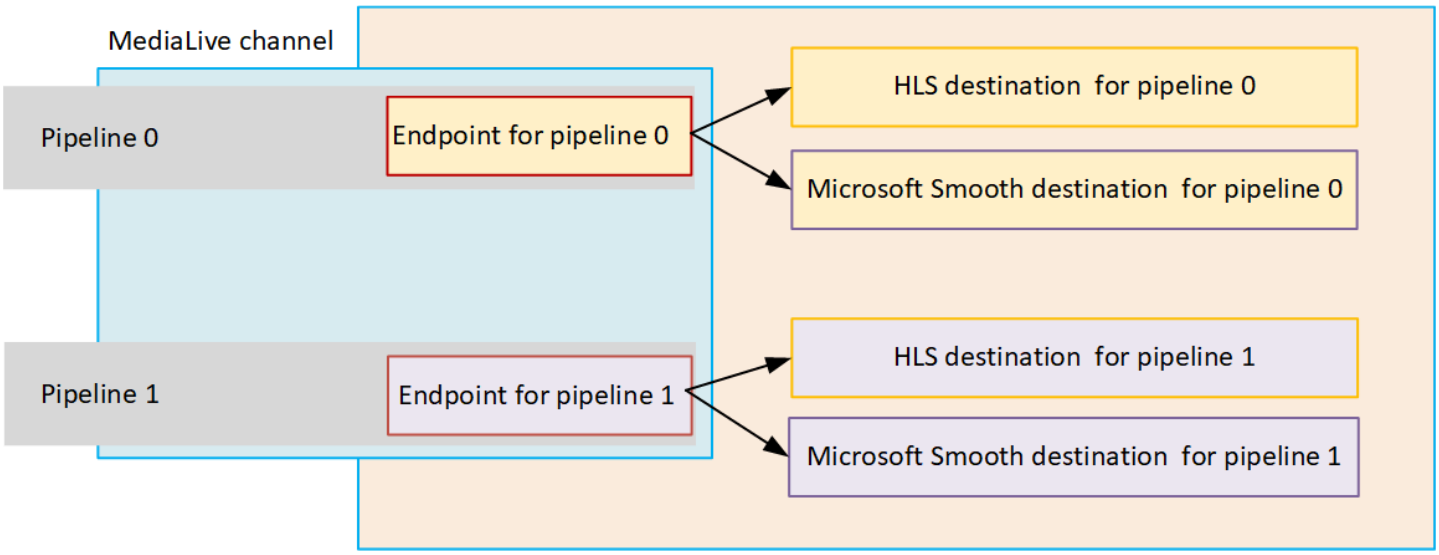
Topics

Use case A – no VPC inputs

This use case applies if the channel won't have inputs that use the VPC:

- No MediaConnect inputs
- No CDI inputs
- No RTMP VPC inputs
- No RTP VPC inputs

Here is a diagram of the setup, when the channel is a standard channel. In this example, the channel has two output groups. Assume that the destinations of both the output groups are on EC2 on your VPC.



Single-pipeline channels

You must identify subnets for the following locations:

- The channel endpoint for pipeline 0 (in the blue box).
- The destinations for pipeline 0 (in the orange box).

Your setup must observe these rules for VPCs and subnets:

- You can set up the locations on any number of VPCs.
- There is no requirement for any of the VPCs or subnets to be the same or different.

Your setup must observe these rules for the Availability Zones of the subnets that you identify:

- The channel endpoint can be in the same Availability Zone as the destination (or destinations)or in a different Availability Zone. If it is in a different Availability Zone, you will incur outgoing data transfer charges. For more information about pricing, see https://aws.amazon.com/medialive/pricing/.

Standard channels

You must identify subnets for the following:

- The two channel endpoints (in the blue box).
- All the destinations (in the orange box).

Your setup must observe these rules for VPCs and subnets:

- You can set up the locations on any number of VPCs.
- The subnets for the channel endpoints must be different from each other, but the two subnets must be on the same VPC.
- There are no other requirements for subnet uniqueness in any of the subnets that you identify.

Your setup must observe these rules for the Availability Zones of the subnets that you identify:

- The Availability Zones for the two channel endpoints must be different.

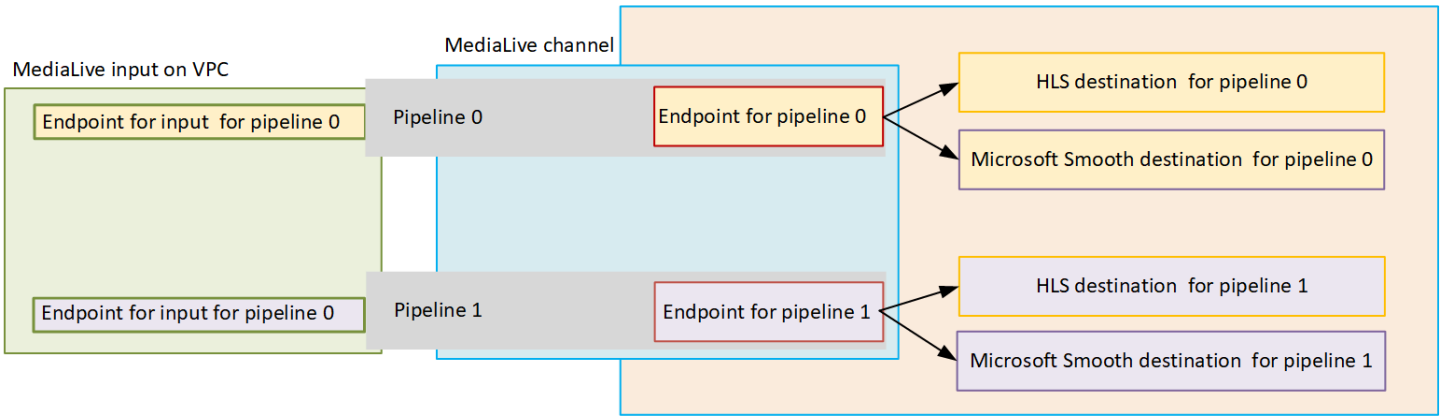
- Each channel endpoint can be in the same Availability Zone as the destination (or destinations). Or it can be in a different Availability Zone. If you choose to set up with different Availability Zones, you will incur outgoing data transfer charges. For more information about pricing, see https://aws.amazon.com/medialive/pricing/.

Use case B – channel includes VPC inputs

This use case applies if the channel includes inputs that use the VPC:

- MediaConnect inputs
- CDI inputs
- RTMP VPC inputs
- RTP VPC inputs

Here is a diagram of the setup, when the channel is a standard channel. In this example, the channel has at least one VPC input. It also has two output groups. Assume that the destinations of both the output groups are on EC2 on your VPC.



Single-pipeline channels

You must identify subnets for the following locations:

- The endpoint for the VPC input for pipeline 0 (in the green box).
- The channel endpoint for pipeline 0 (in the blue box).
- The destinations for pipeline 0 (in the orange box).

Your setup must observe these rules for VPCs and subnets:

- You can set up the locations on any number of VPCs.
- There is no requirement for any of the VPCs or subnets to be the same or different.

Your setup must observe these rules for the Availability Zones of the subnets that you identify:

- The endpoint of the VPC input and the channel endpoint must be in the same Availability Zone. This rule exists because both these endpoints are inside the channel pipeline, and the pipeline can't start in one Availability Zone and end in another.

  If the VPC input is already set up in the VPC, it is probably easiest to identify the Availability Zone of that subnet as the shared Availability Zone.

  If the VPC input isn't yet set up, make sure that the two subnets are in the same Availability Zone.
- The channel endpoint can be in the same Availability Zone as the destination (or destinations) or in a different Availability Zone. If it is in a different Availability Zone, you will incur outgoing data transfer charges. For more information about pricing, see https://aws.amazon.com/medialive/pricing/.

Standard channels

You must identify subnets for the following:

- The endpoints for the VPC inputs (in the green box).
- The channel endpoints (in the blue box).
- The destinations (in the orange box).

Your setup must observe these rules for VPCs and subnets:

- You can set up the locations on any number of VPCs.
- The subnet for the VPC inputs in pipeline 0 and the VPC inputs in pipeline 1 must be on the same VPC. They can be on the same or different subnets.
- The subnet for the channel endpoint in pipeline 0 and the channel endpoint in pipeline 1 must be different from each other, but the two subnets must be on the same VPC.
- There are no other requirements for subnet uniqueness in any of the VPCs or subnets that you identify.

Your setup must observe these rules for Availability Zone:

- The Availability Zones for the two channel endpoints must be different.
- Within each pipeline, the endpoint of the VPC input and the channel endpoint must be in the same Availability Zone. This rule exists because both these endpoints are inside the channel pipeline, and the pipeline can't start in one Availability Zone and end in another.

  If the VPC input is already set up in the VPC, it is probably easiest to identify the Availability Zone of that subnet as the shared Availability Zone.

  If the VPC input isn't yet set up, make sure that the subnets are in the same Availability Zone.
- Within each pipeline, each channel endpoint can be in the same Availability Zone as the destination (or destinations). Or it can be in a different Availability Zone. If you choose to set up with different Availability Zones, you will incur outgoing data transfer charges. For more information about pricing, see https://aws.amazon.com/medialive/pricing/.

Working with AWS Elemental MediaLive resources

This section provides detailed instructions about how to create, view, edit and delete MediaLive resources: channels, devices, inputs, input security groups, and multiplexes.

Topics

Working with AWS Elemental MediaLive channels

A channel ingests and transcodes (decodes and encodes) source content from the inputs that are attached to that channel, and packages the new content into outputs. You create and configure the channel with the details that instruct the channel how to perform this processing. You then run the channel to start processing.

There are three ways to create a channel:

- **From scratch.** The **Create** form on the MediaLive console contains some fields that display system defaults and other fields that are empty. You can create a channel from scratch by modifying the system defaults and by completing the appropriate empty fields. For more information, see *Setup: Creating a channel* .
- **Using a built-in template or custom template.** You can use a template to create a channel, and reuse the template to create more channels. For more information, see the section called "Creating a channel from a template" later in this chapter.
- **By cloning an existing channel.** You can clone an existing channel, and then edit the settings for the new (cloned) channel. For more information, see the section called "Creating a channel by cloning" later in this chapter.

After you have created the channel, you edit or delete it in the same way, regardless of which method you used to create it.

Topics

Creating a channel from scratch

For information about creating a channel from scratch, see *Setup: Creating a channel* .

Creating a channel from a template

You can create a channel by using a custom template or by using one of the built-in templates that MediaLive provides.

Topics

Using built-in templates

MediaLive includes built-in templates that you can access on the console. Each template includes data for output groups and outputs, and most importantly, data for encoding video to meet specific use cases (as specified in the template description).

When you use a built-in template, all sections of the **Create channel** page are populated with data except for the inputs and output destinations sections.

Even though the templates are built-in, you can choose to edit the existing fields and complete the empty fields.

Using custom templates

You or another person in your organization may have created custom templates. A custom template might contain nearly all the data that is required to create a complete channel, or it might contain only portions of the data. To create a custom template, see the section called “Creating a custom template”.

Typically, templates are created in order to be shared among different users.

If your organization uses templates, you must obtain the templates you will use from the person who created the templates. You must store them in a folder on the computer where you are working on the MediaLive console. This folder is the "custom template location." You perform this task in your computer's filesystem, outside of MediaLive.

When you use a custom template, MediaLive populates all sections of the **Create channel** page with data from the template, except for the input data. Even if the template includes input data, that data will not be pulled into the **Create channel** page.

You can edit the existing fields and complete the empty fields as needed.

Creating a channel from a template

To create a channel from a template (console)

1. If you plan to use a custom template, make sure you have set up to use them. See the section called “Using custom templates”.
2. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
3. In the navigation pane, choose **Channels**. On the **Channels** page, choose **Create channel**.
4. On the **Create channel** page, in the **Channel and input details** section, in the **Channel template** section, do one of the following:
   - To use a built-in template: For **Template**, from the **Channel templates** section of the drop-down list, choose a template. (The **Existing channels** section does not list templates.)
   - To use a custom template: Choose **Select custom template**. Navigate to the "custom template" folder and choose the template. For information on the custom template location, see the section called “Using custom templates”.
5. Complete the fields, such as the input fields, that must always be completed. You can also edit other fields as needed. For more information, see *Setup: Creating a channel* .

Creating a custom template

You create a custom template by exporting the data from an existing (and therefore validated) channel. MediaLive exports the data to a JSON file that you can use on the console .

To create a custom template (console)

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Channels**. On the **Channels** page, choose the channel name (not the radio button).
3. In **Channel actions**, choose **Download custom template**. Follow the prompts to save the channel as a template. The template is a JSON file with the same name as the channel.
4. (Optional) Open the file in a suitable editor and make changes. For example, you can change field values, add fields, and remove fields. Be careful to maintain valid JSON.

   You don't need to remove the input attachments. When you use this template in a new channel, MediaLive imports all the data except for the input attachments.
5. Make the custom template available to the users who will need them. Each user must store the template in a folder that is accessible from the computer where the user will work on the MediaLive console. This task is performed outside of MediaLive.

Users can use the template file on the MediaLive console.

Creating a channel by cloning

Cloning lets you use an existing channel as the basis for a new channel. When you clone an existing channel, all sections of the **Create channel** page are populated with the data from the cloned channel, *except* for the following:

- The input sections. These sections are always empty in the cloned channel.
- The tags. There are no tags in the cloned channel.

You can edit the existing fields and complete the empty fields as needed.

You can clone a channel that is in the **Channels** list. (You can also clone a channel after choosing **Create channel**; for more information, see the section called "Creating a channel from a template".)

To create a channel by cloning (console)

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Channels**.
3. On the **Channels** page, choose the radio button next to the channel name.
4. Choose **Clone**.

   The **Create channel** page appears with all the original data except for the inputs and the tags.
5. Give the channel a new name and complete the input sections. Change other fields as needed. For more information, see *Setup: Creating a channel* .

Editing and deleting a channel

You can edit an existing (saved) channel to change how it processes the input, and you can delete a channel. However, you can edit or delete a channel only when it is not running.

Editing a channel

You can edit any existing channel by editing, adding, or deleting output groups and outputs. you can also edit, add, or delete the channel's video, audio, and caption encodes.

The channel must be idle (not running).

> **Note**
>
> You can't change the class for a channel by editing the channel. Instead, see the section called "Updating channel class".

To edit a channel

1. On the **channels** page, choose the option by the channel name.

2. Choose actions, and then choose **edit**. The edit channel page appears. The details on this page are identical to those on the **create channel** page. For information about working with this page, see *Setup: Creating a channel* .
3. When done, choose update channel.

   Wait for the channel **state** to return to **idle** before performing another action with this channel.

Editing the tags associated with a channel

You can edit the tags associated with a channel at any time, when the channel is running or when it is idle. You can add more tags (up to the limit), and you can delete tags.

To edit the tags in a channel

1. On the **Channels** page, choose the channel name.
2. Choose the Tags tab. Add or delete tags. To edit the value of an existing tag, delete the tag and add it again. For more information, see the section called “Tagging resources”.
3. When done, choose Save.

Deleting a channel

You can delete a channel from the **Channels** list or the details view.

The channel must be idle (not running).

To delete a channel

1. On the **Channels** page, choose the option by the channel name.
2. If the channel is running, choose **Stop**.
3. Choose **Delete**.

Update the channel class—pipeline redundancy

You can change the channel class of an existing channel in order to enable or disable pipeline redundancy in the channel.

For general information on channel class and its role in the channel, see the section called "Pipeline redundancy".

For the procedure to change the class, see the section called "Changing an existing channel".

Viewing a channel configuration

You can view information about the configuration of a channel on the **Channel details** page on the AWS Elemental MediaLive console. This page is useful for viewing information when the channel is running. (When a channel is running, you can't view details by choosing **Edit**).

To view configuration information (AWS Elemental MediaLive console)

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Channels**. (For information about the buttons on this page, see the section called "Editing a channel", *Operations: Start, stop, and pause channel*, and the section called "Creating a channel by cloning".)
3. To view more details about a channel, choose the name of that channel. The **Channel details** page appears.
4. View configuration information in one of these places:

   - For information about the input specification for the channel, choose the **Details** tab and look at the **Input specifications** pane.
   - For a one-click view of the destination for the channel (on the downstream system), choose the **Destinations** tab.
   - For basic information about the configuration of the channel, choose the **Details** tab.
   - For a read-only view of the complete configuration of the channel (which you specified when you created or edited the channel), choose the **Settings** tab.
   - For a view of the raw JSON code for the channel configuration, choose the **Details** tab, and then choose **Advanced** details. You can copy this JSON code to your clipboard.

Working with Link input device

A Link input device is the interface in MediaLive for an AWS Elemental Link *hardware device* that's connected to MediaLive. For general information about this hardware, see the section called "AWS Elemental Link".

For information about working with the Link input device in MediaLive, see *Setup: AWS Elemental Link* .

Working with inputs in AWS Elemental MediaLive

An *input* is a video asset that is to be transcoded and packaged. The source of the video asset is the upstream system—the system in your end-to-end workflow whose activities occur before those of AWS Elemental MediaLive. The upstream system can be on the public internet or in a virtual private cloud (VPC) that you created using Amazon Virtual Private Cloud (Amazon VPC).

An AWS Elemental MediaLive input holds information that describes how the source content on the upstream system and the MediaLive channel are connected.

Categories for inputs

Inputs can be categorized in several ways:

- **Type** – An input has a type of source and delivery protocol. For example, an HLS input or an RTMP input. For more information, see the section called “Input types”.
- **Live versus VOD** – An input is either a live (streaming) input or a video on demand (VOD) input. For more information, see the section called “Input types”.
- **Push versus pull** – An input is either a push input or a pull input.
  - With a push input, the upstream system pushes the input to *endpoints* on MediaLive. The input holds these endpoints.
  - With a pull input, MediaLive pulls the input from the upstream system. The input holds these *source* addresses on the upstream system.

  For more information, see the section called “Input types”.
- **Input class** – An input can be set up as either a standard-class input or single-class input:
  - You can use a standard-class input with a standard channel or a single-pipeline channel.
  - You can use a single-class input only with a single-pipeline channel.

  For more information on the purpose of input classes, see the section called “Pipeline redundancy”.

  For information on the classes applicable to each input type, see the section called “Supported input class”.

- **Static versus dynamic** – When you create the input, you decide if it is static or dynamic.
  - A static input has a URL (that points to the content source) that never changes.

    Any input type can be set up as a static input.
  - A dynamic input has a URL that includes a variable portion. It is intended for use with input switching.

    Only MP4 and Transport Stream (TS) inputs can be set up as dynamic inputs.

  For more information, see the section called “Input switching”.

Inputs, input security groups, and channels

The input is one of the components of a MediaLive workflow. The others are the input security group and the channel. These three components are linked together. If the input requires it, an input security group is attached to the input. Not all inputs have this requirement. An input is attached to a channel.

The following rules apply to the linking to an input:

- The association between an *input* and an *input security group* is defined in the input side. You set up the association when you create or edit the input.
- The association between an *input* and a *channel* is defined on the channel side. You set up the association when you create or edit the channel.
- An input can have only one input security group attached to it. But that input security group can be already attached to another input; one input security group can serve several inputs.
- An input can be attached to only one channel; several channels can't use the same input.

Creating an input

For information about creating an input, see Creating a channel from scratch.

Editing an input

The rules for editing an input are as follows.

Changing the input security group

- You can attach a different input security group.

Changing the endpoint (push input) or sources (pull input)

- For an RTP input or an RTMP push input that isn't for a VPC, you can edit the fields in the input endpoint.
- For an RTP VPC input or an RTMP VPC push input, you *can't* edit the IP addresses for the input endpoint. To change these addresses, you must delete the input and create it again.
- For an Elemental Link input, you can attach a different AWS Elemental Link.
- For a MediaConnect push input, you can edit the ARNs to refer to different AWS Elemental MediaConnect flows. The outputs for the former ARNs will be deleted in MediaConnect, and new outputs (with new IDs) for the new ARNs will be created.
- For a pull input, you can edit the fields in an input source.

Changing the input class

- You can't change the input class if the input is attached to a channel. For more information about changing the class of inputs and channels, see the section called “Changing an existing channel”.

Changing input type

- You can't change the type of an input. For example, if you set up an input as an RTMP push but it is actually an HLS input, delete the input and create it again.

Rules for the state of the input and channel

There are constraints on performing these edits, as follows:

- If an input is attached to a channel, you can edit the input only if the channel is idle.
- If an input is attached to a channel and an input security group, you can edit the input only if the channel is idle.
- If an input is not attached to a channel, you can edit it at any time, even if it is attached to an input security group.

To edit an input

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Inputs**. Choose the name of the input, and then choose **Edit**.
3. On the **Inputs** page, make the following changes as appropriate:
   - You can change the **Name**.
   - You can't change the **Input type**. If the input has the wrong type, delete it and create it over again.
   - You can change the **Input devices** (applies only to an Elemental Link input).
   - You can change the **Input class** only if the input isn't attached to a channel. For more information, see the section called “Changing an existing channel”.
   - You can change the **Source** section (applies only to pull inputs).
   - You can change the **Endpoint** section only on an RTP input or RTMP push input that isn't for a VPC.
   - You can change the **Input security groups** section (applies only to push inputs that aren't for a VPC).
   - In the **Tags** section, you can add or delete tags. To edit the value of an existing tag, delete the tag and add it again. For more information, see the section called “Tagging resources”.
4. Choose **Update**.

   Wait for the input **State** to return to **In use** or **Idle** before performing another action with this input.

Deleting an input

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Inputs**. On the **Inputs** page, find the input that you want to delete, and then look at the **State** column.
   - If the state is **Detached**, then choose the input and choose**Delete**.
   - If the state is **Attached** and you want to delete the input but keep the channel, first detach the input. Then come back to this **Inputs** page, choose the input, and choose**Delete**.
   - If the state is **Attached** and you want to delete both the input and its channel, then first delete the channel. Then come back to this **Inputs** page, choose the input, and choose**Delete**.

The results are as follows:

- If the input is an Elemental Link input, MediaLive deletes the input. But the Link input device remains in the **Devices** list, and you can attach it to a new input at any time.
- If the input is a MediaConnect push input, the corresponding outputs in MediaConnect are automatically deleted. You don't have to delete the outputs.
- If the input is an RTP VPC input or an RTMP VPC push input, the elastic network interfaces of the endpoints are deleted and the IPv4 addresses in the subnet are released for use by another resource. You don't have to delete the network interfaces.

The input security group that is attached to the input (if any) is *not* deleted.

Detaching an input

You can detach an input from a channel. The channel must be idle.

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Inputs**. Find the input in the list and choose its name. On the **Input Details**, find the ID of the channel that the input is attached to.
3. Choose that ID. The **Channel details** page for that channel appears.
4. Choose **Channel actions**, then choose **Edit channel**.
5. In the list of input attachments on the left, find the the name of the input to detach. Choose the name.
6. In the **Input attachment details** panel, choose **Remove**. The input is detached.
7. Choose **Update channel** at the bottom of the page.

Working with input security groups

An *input security group* contains a list of rules. Each rule is a range of IP addresses (CIDR blocks) that are allowed to push content to MediaLive. When you attach an input security group to an input, you apply a rule to that input—only an upstream system with an IP address from one of the ranges in that input security group is allowed to push content to that input. MediaLive will ignore push requests from IP addresses not covered by that input security group.

You can include up to 10 rules (IP address ranges or CIDR blocks) in one input security group.

You can attach the same input security group to any number of inputs.

Topics

Purpose of an input security group

Input security groups are used with specific *push* inputs where the upstream system for the source is on the public internet:

- They are used for RTP inputs and RTMP push inputs that don't use a VPC.
- They aren't used for RTP VPC inputs, RTMP VPC push inputs, MediaConnect inputs, or Elemental Link inputs. These inputs implement security in other ways.

An input security group restricts access to the input. The group prevents unauthorized third parties from pushing content from the public internet to an input and to the channel that this input is attached to. Without the protection of this feature, any third party can push content to a MediaLive input if they know the input IP address and port. Note that setting permissions on the account that owns the channel does not prevent this third-party push; only an input security group prevents it.

You can attach an input security group to more than one input. In other words, one input security group can serve several inputs.

Creating an input security group

You create an input security group to specify a list of access rules. When you create a push input, you must attach an input security group, in order to restrict access to the input

You can include up to 10 rules (IP address ranges or CIDR blocks) in one input security group.

You can attach the same input security group to any number of inputs.

To create an input security group

1. Identify the IP addresses that the upstream systems will push from. These IP addresses might be on the public internet or they might be on your LAN or WAN.
2. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
3. In the navigation pane, choose **Input security groups**.
4. On the **Input security groups** page, choose **Create input security group**.
5. For **New security group**, type one or more IPv4 CIDR blocks.

   Each CIDR block must include a subnet mask. In the examples below, the subnet mask is the */nn* portion.

   Separate the entries with commas, or type each entry on a separate line.

   You might not know how to form a CIDR block for your range of IP addresses. If so, search on the internet for "IP CIDR calculator" to find an online converter tool.
6. In the **Tags** section, create tags if you want to associate tags with this input security group. For more information, see the section called “Tagging resources”.
7. Choose **Create**.

Example 1

192.0.2.0/24

This CIDR block covers all IP addresses that start with *192.0.2*

Example 2

192.0.2.111/32

This CIDR block covers the single IP address *192.0.2.111*

Editing an input security group

You can edit any of the fields in an input security group. You can perform these edits at any time, even if the input security group is attached to an input that is attached to a channel that is running.

To edit an input security group

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Input Security Groups**.
3. On the **Input security groups** page, choose the input security group, and then choose **Edit**.
4. Change any fields as appropriate, and then choose **Update**.

   Wait for the input security **State** to return to **In use** or **Idle** before performing another action with this input security group.

To add, delete, or edit tags in an input security group

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Input Security Groups**.
3. On the **Input security groups** page, choose the name of the input security group. Do not choose **Edit**.
4. On the **Input security group** page for this input security group, in the **Tags** section, add or delete tags. To edit the value of an existing tag, delete the tag and add it again. For more information, see the section called “Tagging resources”.

   Wait for the input security **State** to return to **In use** or **Idle** before performing another action with this input security group.

Deleting an input security group

You can delete an input security group so long as it is not attached to any inputs.

To delete an input security group

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Input security groups**.
3. On the **Input security groups** page, look at the **State** for the group to delete:
   - If the **State** is **Idle**, choose the group, and then choose **Delete**.
   - If the **State** is **In use**, continue with this procedure.

4. Make a note of the ID of the input security group. For example, 1234567.
5. Choose the group, and then choose **Edit**.
6. On the **Edit input security group** page, look at the **Inputs** on the right side and count how many inputs are attached to this input security group.
7. Choose the first input. Then on the page for that input, choose **Edit**. On the **Edit** page, in the **Input security group**, either create a new input security group for this input or choose another group (make sure you don't rechoose the same group; check the ID that you noted earlier). Choose **Update** so that this input is no longer attached to the input security group that you want to delete.
8. If there are still more inputs associated with this input group, then in the navigation pane, choose **Input security groups**, and repeat these steps to detach this input security group from all the inputs.
9. After detaching the last input from this input security group, wait for the **State** of the input security group to specify **Idle**. Then choose the group, and choose **Delete**.

Working with MediaLive multiplexes

A MediaLive multiplex creates a multi-program transport stream (MPTS). You might be interested in creating a MediaLive multiplex if you are a service provider who has experience with distributing transport stream (TS) content over RTP or UDP.

To set up a multiplex, you create a MediaLive multiplex. You then add MediaLive programs to the multiplex. Finally, you create one MediaLive channel for each program, and associate each channel with its program.

For conceptual information about setting up a multiplex, see the section called “Multiplex and MPTS”.

Topics

Summary of actions

The following table summarizes the create, edit, and delete capabilities for the multiplex, program, and channel.

Item	Action	Note
Multiplex	Create	
	Edit	The multiplex can be idle or running. The channels can be all idle, or all running, or a combination or idle and running. Exception: To change the **Max Video Buffer Delay** field, the multiplex must be idle.
	Delete	The multiplex must be idle, and must not have any associated programs.
Program	Create	The multiplex for the program can be idle or running.
	Edit	The multiplex for this program can be idle or running. The channel for this program can be idle or running.
	Delete	The multiplex for this program can be idle or running. The program can't have any associated channel.

Item	Action	Note
Channel	Create	The multiplex for this channel can be idle or running. The program for the channel must be empty.
	Edit	The channel must be idle. The multiplex for this channel can be idle or running.
	Delete	The channel must be idle. The channel can still be attached to a program.

Creating a multiplex and program

A MediaLive multiplex provides configuration information for an MPTS, including the bitrate of the entire MPTS.

You can create a multiplex from scratch, or you can clone an existing multiplex. Cloning a multiplex is similar to cloning a channel—the values in most of the fields are copied to the new multiplex.

You can create a program inside a multiplex. You can't create a program without attaching it to a multiplex.

To create a multiplex

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Multiplexes**.
3. On the **Multiplexes** page, choose **Create**.
4. Complete the fields on the **Create multiplex** page.
5. Choose **Create**.

   The multiplex is added to the **Multiplexes** page. After the status of the multiplex changes to IDLE, your next step is to add programs to the multiplex. For more information, see later in this section.

To create a multiplex by cloning

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Multiplexes**, and then choose the multiplex that you want to clone.
3. On the **Details** pane, choose **Multiplex actions**, and then choose **Clone**.

To create a program

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Multiplexes**, and then choose the multiplex where you want to add the program.
3. On the **Details** pane, choose the **Programs** tab.
4. Choose **Create program**.
5. Complete the fields on the **Create program** page.
6. Choose **Create**.

   The **Program** details pane appears for this program. Note that the channel state always specifies CHANNEL MISSING.
7. Either now or later, you must create a channel for this program:

   - You can choose **Create channel** to immediately add a channel to this program.
   - You can add a channel later, in the same way that you create a channel that isn't part of a multiplex.

The channel attached to a program is a regular channel in which the output group is always a multiplex output group.

For information about special steps for completing the fields in a channel in a multiplex, see the section called “Step 5: Create the channels”.

Creating a channel

The channel attached to a program is a regular channel in which the output group is always a multiplex output group.

In a new multiplex, you can create a channel as soon as its program has been successfully created.

If a multiplex is running, you don't have to stop the multiplex to add a channel. You can add a channel to a running multiplex.

For information about special steps for completing the fields in a channel in a multiplex, see the section called “Step 5: Create the channels”.

Editing multiplexes, programs, and channels

You can edit a multiplex, the programs in a multiplex, and the channels in a multiplex. There are specific rules that are based on the state (running or idle) of the item that you want to work with, as described in this section.

Editing a multiplex

There are very few restrictions on your ability to edit a multiplex. You can edit a multiplex when these situations apply:

- While the multiplex is idle or running, unless you want to change the **Maximum Video Buffer Delay** field. To change that field, the multiplex must be idle.
- While the channels in the multiplex programs are idle or running.
- While MediaLive is in the process of adding programs that you just created.

To edit a multiplex

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Multiplexes**, and then choose the multiplex that you want to edit.
3. On the **Details** pane, choose **Multiplex actions**, and then choose **Edit**.
4. Make the changes that you want, and then choose **Save changes**.

Editing a program

You can edit a program at any time, including when the multiplex if running or when the associated channel is running.

To edit a program

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.

2. In the navigation pane, choose **Multiplexes**, and then choose the multiplex that you want to edit.
3. On the **Details** pane, choose the **Programs** tab.
4. Choose **Program actions**, and then choose **Edit**.
5. Make the changes that you want, and then choose **Save changes**.

Editing a channel in a program

You can edit a channel that is idle.

To edit a channel

1. Stop the channel. You can stop the channel in the usual way, from the **Channels** pane. Or you can stop it from the **Multiplex** page. For more information, see the section called “Stopping a channel in a multiplex”.
2. Edit the channel. For more information, see the section called “Editing a channel”.

Deleting multiplexes, programs, and channels

You can delete a multiplex, the programs in a multiplex, and the channels in a multiplex. There are specific rules that are based on the state of the item that you want to work with, as described in this section.

Deleting a multiplex

To delete a multiplex, the multiplex must be idle, and all of its programs must be empty (they must not have associated channels.)

To delete a multiplex

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Multiplexes**, and then choose the multiplex that you want to delete.
3. On the **Details** pane, choose **Multiplex actions**, and then choose **Stop**.
4. On the **Programs** pane, choose the first program with a running channel, choose **Program actions**, and then choose **Stop channel**.
5. Repeat for all the channels that are running.

6. Make a note of the names of the channels, and then display the **Channels** page. Choose the channels, choose **Actions**, and then choose **Delete**.
7. Return to the **Multiplex** page.
8. Choose **Multiplex actions**, and then choose **Delete multiplex**. MediaLive deletes the multiplex and all of its programs.

Deleting a program

You can delete a program that has no channel. You can delete a program when the multiplex is running or idle.

To delete a program

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Multiplexes**, and then choose the multiplex that you want to work with.
3. On the **Programs** pane, choose the program to delete.
4. If the channel for that program is running, choose **Program Actions**, and then choose **Stop channel**.
5. Wait for the channel to change to **Idle**.
6. Make a note of the name of the channel, and then display the **Channels** page. Choose the channel, choose **Actions**, and then choose **Delete**.
7. Return to the **Multiplex** page.
8. Choose **Program actions**, and then choose **Delete program**.

Deleting a channel

You can delete a channel when the multiplex is running or idle. You don't detach the channel from its program—there is no concept of detaching a channel from a program.

To delete a channel, display the **Channel** page, and delete the channel in the usual way. For more information, see the section called "Deleting a channel".

Security in AWS Elemental MediaLive

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS compliance programs. To learn about the compliance programs that apply to AWS Elemental MediaLive, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using MediaLive. The following topics show you how to configure MediaLive to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your MediaLive resources.

Topics

Data protection in AWS Elemental MediaLive

The AWS shared responsibility model applies to data protection in AWS Elemental MediaLive. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on

this infrastructure. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-2.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with MediaLive or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

If you provide a URL to an external server, MediaLive requires that you do not include credentials information in the URL to validate your request to that server. If a URL to an external server requires credentials, we recommend that you use the Parameter Store feature in AWS Systems Manager. For more information, and the steps to implement AWS Systems Manager Parameter Store, see Requirements for AWS Systems Manager—creating password parameters in Parameter Store.

AWS Elemental MediaLive doesn't require that you supply any customer data. There are no fields in channels, devices, inputs, input security groups, multiplexes, or reservations, where there is an expectation that you will provide customer data.

MediaLive includes features such as the AWS Systems Manager Parameter Store that provide you with a secure way to handle sensitive information. You should always use these features to pass a password; you should not circumvent them by including a password in a URL.

Deleting data in AWS Elemental MediaLive

You can delete data from AWS Elemental MediaLive by deleting the object, for example, the channel or input. You can delete data using the console, REST API, AWS CLI, or AWS SDKs. The data will be deleted; no further steps are required after you delete data by completing a delete operation.

To delete data using the console, see the following sections:

- the section called “Deleting a channel”
- the section called “Link input device”
- the section called “Deleting an input”
- the section called “Deleting an input security group”
- the section called “ Deleting multiplexes, programs, and channels ”
- the section called “Deleting an expired reservation”

Identity and Access Management for AWS Elemental MediaLive

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use MediaLive resources. IAM is an AWS service that you can use with no additional charge.

Topics

- Audience
- Authenticating with identities
- Managing access using policies
- How AWS Elemental MediaLive works with IAM
- Identity-based policy examples for AWS Elemental MediaLive
- Troubleshooting AWS Elemental MediaLive identity and access

Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in MediaLive.

Service user – If you use the MediaLive service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more MediaLive features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator. If you cannot access a feature in MediaLive, see Troubleshooting AWS Elemental MediaLive identity and access.

Service administrator – If you're in charge of MediaLive resources at your company, you probably have full access to MediaLive. It's your job to determine which MediaLive features and resources your service users should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with MediaLive, see How AWS Elemental MediaLive works with IAM.

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to MediaLive. To view example MediaLive identity-based policies that you can use in IAM, see Identity-based policy examples for AWS Elemental MediaLive.

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. You must be *authenticated* (signed in to AWS) as the AWS account root user, as an IAM user, or by assuming an IAM role.

You can sign in to AWS as a federated identity by using credentials provided through an identity source. AWS IAM Identity Center (IAM Identity Center) users, your company's single sign-on authentication, and your Google or Facebook credentials are examples of federated identities. When you sign in as a federated identity, your administrator previously set up identity federation using IAM roles. When you access AWS by using federation, you are indirectly assuming a role.

Depending on the type of user you are, you can sign in to the AWS Management Console or the AWS access portal. For more information about signing in to AWS, see How to sign in to your AWS account in the *AWS Sign-In User Guide*.

If you access AWS programmatically, AWS provides a software development kit (SDK) and a command line interface (CLI) to cryptographically sign your requests by using your credentials. If

you don't use AWS tools, you must sign requests yourself. For more information about using the recommended method to sign requests yourself, see Signing AWS API requests in the *IAM User Guide*.

Regardless of the authentication method that you use, you might be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see Multi-factor authentication in the *AWS IAM Identity Center User Guide* and Using multi-factor authentication (MFA) in AWS in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you don't use the root user for your everyday tasks. Safeguard your root user credentials and use them to perform the tasks that only the root user can perform. For the complete list of tasks that require you to sign in as the root user, see Tasks that require root user credentials in the *IAM User Guide*.

Federated identity

As a best practice, require human users, including users that require administrator access, to use federation with an identity provider to access AWS services by using temporary credentials.

A *federated identity* is a user from your enterprise user directory, a web identity provider, the AWS Directory Service, the Identity Center directory, or any user that accesses AWS services by using credentials provided through an identity source. When federated identities access AWS accounts, they assume roles, and the roles provide temporary credentials.

For centralized access management, we recommend that you use AWS IAM Identity Center. You can create users and groups in IAM Identity Center, or you can connect and synchronize to a set of users and groups in your own identity source for use across all your AWS accounts and applications. For information about IAM Identity Center, see What is IAM Identity Center? in the *AWS IAM Identity Center User Guide*.

IAM users and groups

An *IAM user* is an identity within your AWS account that has specific permissions for a single person or application. Where possible, we recommend relying on temporary credentials instead of creating

IAM users who have long-term credentials such as passwords and access keys. However, if you have specific use cases that require long-term credentials with IAM users, we recommend that you rotate access keys. For more information, see Rotate access keys regularly for use cases that require long-term credentials in the *IAM User Guide*.

An *IAM group* is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named *IAMAdmins* and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see When to create an IAM user (instead of a role) in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity within your AWS account that has specific permissions. It is similar to an IAM user, but is not associated with a specific person. You can temporarily assume an IAM role in the AWS Management Console by switching roles. You can assume a role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see Using IAM roles in the *IAM User Guide*.

IAM roles with temporary credentials are useful in the following situations:

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see Creating a role for a third-party Identity Provider in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permissions sets, see Permission sets in the *AWS IAM Identity Center User Guide*.
- **Temporary IAM user permissions** – An IAM user or role can assume an IAM role to temporarily take on different permissions for a specific task.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource

(instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.

- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
  - **Forward access sessions (FAS)** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.
  - **Service role** – A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.
  - **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.
- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Using an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

To learn whether to use IAM roles or IAM users, see When to create an IAM role (instead of a user) in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when a principal (user, root user, or role session) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

By default, users and roles have no permissions. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Creating IAM policies in the *IAM User Guide*.

Identity-based policies can be further categorized as *inline policies* or *managed policies*. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, see Choosing between managed policies and inline policies in the *IAM User Guide*.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that

support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Access control lists (ACLs)

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs. To learn more about ACLs, see Access control list (ACL) overview in the *Amazon Simple Storage Service Developer Guide*.

Other policy types

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of an entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see How SCPs work in the *AWS Organizations User Guide*.

- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see Session policies in the *IAM User Guide*.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the *IAM User Guide*.

How AWS Elemental MediaLive works with IAM

Before you use IAM to manage access to MediaLive, learn what IAM features are available to use with MediaLive.

IAM features you can use with AWS Elemental MediaLive

IAM feature	MediaLive support
Identity-based policies	Yes
Resource-based policies	No
Policy actions	Yes
Policy resources	Yes
Policy condition keys (service-specific)	Yes
ACLs	No
ABAC (tags in policies)	Partial
Temporary credentials	Yes
Principal permissions	Yes
Service roles	Yes

IAM feature	MediaLive support
Service-linked roles	No

To get a high-level view of how MediaLive and other AWS services work with most IAM features, see AWS services that work with IAM in the *IAM User Guide*.

Identity-based policies for MediaLive

Supports identity-based policies	Yes

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Creating IAM policies in the *IAM User Guide*.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. You can't specify the principal in an identity-based policy because it applies to the user or role to which it is attached. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

Identity-based policy examples for MediaLive

To view examples of MediaLive identity-based policies, see Identity-based policy examples for AWS Elemental MediaLive.

Resource-based policies within MediaLive

Supports resource-based policies	No

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified

principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. Adding a cross-account principal to a resource-based policy is only half of establishing the trust relationship. When the principal and the resource are in different AWS accounts, an IAM administrator in the trusted account must also grant the principal entity (user or role) permission to access the resource. They grant permission by attaching an identity-based policy to the entity. However, if a resource-based policy grants access to a principal in the same account, no additional identity-based policy is required. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

Policy actions for MediaLive

Supports policy actions	Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Policy actions usually have the same name as the associated AWS API operation. There are some exceptions, such as *permission-only actions* that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

Include actions in a policy to grant permissions to perform the associated operation.

For a list of actions in MediaLive and in other services that users of might need access to when working with MediaLive, see the section called “Reference: summary of user access”.

Policy actions in MediaLive use the following prefix before the action:

```
medialive
```

To specify multiple actions in a single statement, separate them with commas.

```
"Action": [
```

```
      "medialive:action1",
      "medialive:action2"
         ]
```

To view examples of MediaLive identity-based policies, see Identity-based policy examples for AWS Elemental MediaLive.

Policy resources for MediaLive

Supports policy resources	Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. Statements must include either a `Resource` or a `NotResource` element. As a best practice, specify a resource using its Amazon Resource Name (ARN). You can do this for actions that support a specific resource type, known as *resource-level permissions*.

For actions that don't support resource-level permissions, such as listing operations, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

To see a list of MediaLive resource types and their ARNs, see Resources defined by AWS Elemental MediaLive in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see Actions defined by AWS Elemental MediaLive.

To view examples of MediaLive identity-based policies, see Identity-based policy examples for AWS Elemental MediaLive.

Policy condition keys for MediaLive

Supports service-specific policy condition keys	Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Condition` element (or `Condition` *block*) lets you specify conditions in which a statement is in effect. The `Condition` element is optional. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request.

If you specify multiple `Condition` elements in a statement, or multiple keys in a single `Condition` element, AWS evaluates them using a logical `AND` operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical `OR` operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an IAM user permission to access a resource only if it is tagged with their IAM user name. For more information, see IAM policy elements: variables and tags in the *IAM User Guide*.

AWS supports global condition keys and service-specific condition keys. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

To see a list of MediaLive condition keys, see Condition keys for AWS Elemental MediaLive in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see Actions defined by AWS Elemental MediaLive.

To view examples of MediaLive identity-based policies, see Identity-based policy examples for AWS Elemental MediaLive.

ACLs in MediaLive

Supports ACLs	No

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

MediaLive doesn't support ACLs, which means that you don't create ACLs within MediaLive.

However, MediaLive does include a feature that lets you transfer ownership of content that MediaLive stores in an Amazon S3 bucket. You can transfer ownership by setting up MediaLive to include a specific access control list (ACL) when delivering to the bucket. The owner of the bucket creates the ACL, then provides you with the ACL to use. For more information about ACLs, see the section called “Amazon S3 access control lists (ACLs)”.

ABAC with MediaLive

Supports ABAC (tags in policies)	Partial

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes. In AWS, these attributes are called *tags*. You can attach tags to IAM entities (users or roles) and to many AWS resources. Tagging entities and resources is the first step of ABAC. Then you design ABAC policies to allow operations when the principal's tag matches the tag on the resource that they are trying to access.

ABAC is helpful in environments that are growing rapidly and helps with situations where policy management becomes cumbersome.

To control access based on tags, you provide tag information in the condition element of a policy using the `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

If a service supports all three condition keys for every resource type, then the value is **Yes** for the service. If a service supports all three condition keys for only some resource types, then the value is **Partial**.

For more information about ABAC, see What is ABAC? in the *IAM User Guide*. To view a tutorial with steps for setting up ABAC, see Use attribute-based access control (ABAC) in the *IAM User Guide*.

Using temporary credentials with MediaLive

Supports temporary credentials	Yes

Some AWS services don't work when you sign in using temporary credentials. For additional information, including which AWS services work with temporary credentials, see AWS services that work with IAM in the *IAM User Guide*.

You are using temporary credentials if you sign in to the AWS Management Console using any method except a user name and password. For example, when you access AWS using your company's single sign-on (SSO) link, that process automatically creates temporary credentials. You also automatically create temporary credentials when you sign in to the console as a user and then

switch roles. For more information about switching roles, see Switching to a role (console) in the *IAM User Guide*.

You can manually create temporary credentials using the AWS CLI or AWS API. You can then use those temporary credentials to access AWS. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM.

Cross-service principal permissions for MediaLive

Supports forward access sessions (FAS)	Yes

When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.

An example of an action that requires permissions in both MediaLive and another service is creation of a password parameter using the MediaLive console. The console user (the principal) needs permission to create a channel. They also need permission for the **PutParameter** action in the AWS Systems Manager.

For a list of actions in other services that users of might need access to when working with MediaLive, see the section called “Reference: summary of user access”.

Service roles for MediaLive

Supports service roles	Yes

A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.

> **Warning**
>
> Changing the permissions for a service role might break MediaLive functionality. Edit service roles only when MediaLive provides guidance to do so.

Service-linked roles for MediaLive

Supports service-linked roles	No

A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

For details about creating or managing service-linked roles, see AWS services that work with IAM. Find a service in the table that includes a `Yes` in the **Service-linked role** column. Choose the **Yes** link to view the service-linked role documentation for that service.

Identity-based policy examples for AWS Elemental MediaLive

By default, users and roles don't have permission to create or modify MediaLive resources. They also can't perform tasks by using the AWS Management Console, AWS Command Line Interface (AWS CLI), or AWS API. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

To learn how to create an IAM identity-based policy by using these example JSON policy documents, see Creating IAM policies in the *IAM User Guide*.

For details about actions and resource types defined by MediaLive, including the format of the ARNs for each of the resource types, see Actions, resources, and condition keys for AWS Elemental MediaLive in the *Service Authorization Reference*.

Topics

- Policy best practices
- Using the MediaLive console

- Allow users to view their own permissions

Policy best practices

Identity-based policies determine whether someone can create, access, or delete MediaLive resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as AWS CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see IAM Access Analyzer policy validation in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Configuring MFA-protected API access in the *IAM User Guide*.

For more information about best practices in IAM, see Security best practices in IAM in the *IAM User Guide*.

Using the MediaLive console

To access the AWS Elemental MediaLive console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the MediaLive resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won't function as intended for entities (users or roles) with that policy.

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that they're trying to perform.

To ensure that users and roles can still use the MediaLive console, also attach the MediaLive *ConsoleAccess* or *ReadOnly* AWS managed policy to the entities. For more information, see Adding permissions to a user in the *IAM User Guide*.

Allow users to view their own permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
```

```
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Troubleshooting AWS Elemental MediaLive identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with MediaLive and IAM.

Topics

- I am not authorized to perform an action in MediaLive
- I am not authorized to perform iam:PassRole
- I want to allow people outside of my AWS account to access my MediaLive resources

I am not authorized to perform an action in MediaLive

If you receive an error that you're not authorized to perform an action, your policies must be updated to allow you to perform the action.

The following example error occurs when the `mateojackson` IAM user tries to use the console to view details about a fictional *`my-example-widget`* resource but doesn't have the fictional `medialive:`*`GetWidget`* permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 medialive:GetWidget on resource: my-example-widget
```

In this case, the policy for the `mateojackson` user must be updated to allow access to the *`my-example-widget`* resource by using the `medialive:GetWidget` action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, your policies must be updated to allow you to pass a role to MediaLive.

Some AWS services allow you to pass an existing role to that service instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named `marymajor` tries to use the console to perform an action in MediaLive. However, the action requires the service to have permissions that are granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In this case, Mary's policies must be updated to allow her to perform the `iam:PassRole` action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I want to allow people outside of my AWS account to access my MediaLive resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether MediaLive supports these features, see How AWS Elemental MediaLive works with IAM.
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the *IAM User Guide*.

- To learn how to provide access to your resources to third-party AWS accounts, see Providing access to AWS accounts owned by third parties in the *IAM User Guide*.
- To learn how to provide access through identity federation, see Providing access to externally authenticated users (identity federation) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.

Compliance validation for AWS Elemental MediaLive

To learn whether an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program and choose the compliance program that you are interested in. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security and Compliance Quick Start Guides – These deployment guides discuss architectural considerations and provide steps for deploying baseline environments on AWS that are security and compliance focused.
- Architecting for HIPAA Security and Compliance on Amazon Web Services – This whitepaper describes how companies can use AWS to create HIPAA-eligible applications.

> **Note**
>
> Not all AWS services are HIPAA eligible. For more information, see the HIPAA Eligible Services Reference.

- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- AWS Customer Compliance Guides – Understand the shared responsibility model through the lens of compliance. The guides summarize the best practices for securing AWS services and map the guidance to security controls across multiple frameworks (including National Institute of

Standards and Technology (NIST), Payment Card Industry Security Standards Council (PCI), and International Organization for Standardization (ISO)).

- Evaluating Resources with Rules in the *AWS Config Developer Guide* – The AWS Config service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS. Security Hub uses security controls to evaluate your AWS resources and to check your compliance against security industry standards and best practices. For a list of supported services and controls, see Security Hub controls reference.
- Amazon GuardDuty – This AWS service detects potential threats to your AWS accounts, workloads, containers, and data by monitoring your environment for suspicious and malicious activities. GuardDuty can help you address various compliance requirements, like PCI DSS, by meeting intrusion detection requirements mandated by certain compliance frameworks.
- AWS Audit Manager – This AWS service helps you continuously audit your AWS usage to simplify how you manage risk and compliance with regulations and industry standards.

Resilience in AWS Elemental MediaLive

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between Availability Zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

Infrastructure security in AWS Elemental MediaLive

As a managed service, AWS Elemental MediaLive is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see AWS Cloud Security. To design your AWS environment using the best practices for infrastructure security, see Infrastructure Protection in *Security Pillar AWS Well-Architected Framework*.

You use AWS published API calls to access MediaLive through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

Document history for user guide

The following table describes the documentation for this release of AWS Elemental MediaLive

- **API version: latest**

Change	Description	Date
Accessibility data in audio	The guide now includes information about how to include accessibility data in audio outputs.	June 25, 2024
Accessibility data in captions	The guide now includes information about how to include accessibility data in captions outputs.	June 25, 2024
Per-output image overlay	The guide now includes a section on the recently added feature to insert an image overlay in specific outputs in the channel. This feature supplements the existing feature that lets you insert an image overlay in every output in every output group in the channel.	June 20, 2024
Segment breaks triggered by SCTE 35 messages.	You can now configure how MediaLive performs segmentation in some types of output group, when a SCTE 35 message is inserted into the output.	June 14, 2024

CMAF Ingest output group	The guide now includes information about the new CMAF Ingest output group. Sections about planning and creating this output group have been updated. Reference information, such as supported video codecs, has been updated. The sections for the following features, which are supported by this output group have been updated: Nielsen ID3 (converting watermarks to ID3), Nielsen watermarks (insertng watermarks).	June 14, 2024
Profile, bit depth, chroma sampling	The guide now includes information about the values in the Profile field that you fill in when you create a video output encode.	May 23, 2024
Encoding schemes for video	The guide now includes information about the supported video encoding schemes (bit depth, chroma sampling, and so on).	May 23, 2024

Workflow monitor	Analyze AWS media services and create signal maps, visualizations of the media workflow, between those services. Use the signal maps to generate monitoring alarms and notifications using CloudWatch, EventBridge, and AWS CloudFormation.	April 11, 2024
MediaLive alerts	We have added more alerts to the list of MediaLive alerts. Previously, some alerts were omitted in error.	January 9, 2024
Color space using 3D LUT files	The guide now includes a section on the recently added feature to convert color space using 3D LUT files for color mapping.	December 13, 2023
Color space	The guide includes a second section about handling color space. It covers converting and passing through color space when you are working with source video that has dependable color space metadata and that doesn't include an unsupported color space.	December 13, 2023

Per-outputs image overlays	The MediaLive schedule now supports the ability to insert static image overlays in specific outputs. Previously, MediaLive supported only global insertion — insertion in every output in every output group.	October 25, 2023
IAM access for thumbnails	The guide has been updated to describe the IAM access you need to set up when enabling the thumbnails feature.	October 25, 2023
IAM access for trusted entity for AWS Elemental Link devices	The section about setting up MediaLive as a trusted entity for a Link device has been revised to clarify that only one operation in Secrets Manager is required.	September 19, 2023
Permissions for an AWS Elemental Link device	The guide has been updated to include information about permissions that the user needs to work with a Link device.	September 11, 2023
AWS Elemental Link device as the source for a MediaConnect flow.	The guide has been updated to include information about how to set up a Link device as the source for a MediaConnect flow. This new functionality is in addition to the existing ability to set up the device as the source for a MediaLive input.	September 11, 2023

Detaching an input	The guide now includes information about how to detach an input from a channel.	September 7, 2023
Passing through KLV metadata	AWS Elemental MediaLive now supports the ability to extract KLV metadata from an input and to pass it through in TS outputs.	August 24, 2023
Device transfer to another Region	The user guide has been updated to describe the existing feature to transfer a device to another AWS Region.	August 14, 2023
Link Availability Zone control	Link devices have added a new feature that lets you associate each device with a specific AWS Availability Zone.	August 1, 2023
SCTE 104 in SMPTE 2038	The section about SMPTE 2038 now includes previously missing information about extracting SCTE 104 messages. MediaLive has supported SCTE 104 messages since support for SMPTE 2038 was first introduced.	July 31, 2023

Permissions for trusted entity for thumbnails	The information about permissions for the trusted entity has been updated to include the permissions that MediaLive needs to store thumbnails in Amazon S3.	July 13, 2023
Thumbnails	The guide now includes information about the new thumbnails feature that lets you view a preview of the current input for the channel.	July 7, 2023
Channel and multiplex metrics with a 1 second period	The CloudWatch metrics for AWS Elemental MediaLive channels and multiplex now support a period as low as 1 second. This enhanceme nt gives you the ability to monitor activity in your channels in real time.	June 26, 2023
Infrastructure security	The information in this section has been revised. Specifically, we now require TLS 1.2 and we recommend TLS 1.3.	June 24, 2023

Epoch locking and SCTE 35	The guide has been updated to describe a constraint with setting up a channel with epoch locking when an HLS or MediaPackage output group includes SCTE 35 messages. This constraint applies only to epoch locking. It doesn't apply to regular pipeline locking.	June 24, 2023
Data protection	The information in this section has been revised. Specifically, we now require TLS 1.2 and we recommend TLS 1.3.	June 24, 2023
Corrections to delivery to AWS Elemental MediaPackage	The information for using HLS output groups to deliver to MediaPackage using the standard MediaPackage API and using the MediaPackage v2 API have both been revised to correct errors, particula rly in the examples for the MediaPackage channel URLs.	May 31, 2023
AWS Elemental Link reorganiz ation	The information about AWS Elemental MediaLive has been reorganized into three main sections, one in Features, one in the setup topics in the navigation bar, and one in the operations topics in the navigation bar.	May 31, 2023

Timecode configuration	The section about timecode has been renamed to timecodes and timestamps. The section has also been revised to clarify how timecode works in MediaLive.	May 22, 2023
Pipeline locking	This section has been revised to clarify some points. Pipeline locking now works with a channel that has multiple inputs (a channel that implements input switching). Inputs must have an embedded timecode. The timecode configuration field in General Configuration for the channel has no effect on pipeline locking.	May 22, 2023
Correction to pipeline locking	This section has been revised to remove incorrect information. Pipeline locking requires an embedded timecode in the input. But you don't need to set the timecode configuration source to embedded. MediaLive always looks for the embedded timecode.	May 22, 2023

Permissions for trusted entity for MediaPackage v2	The information about permissions for the trusted entity has been updated to include the actions required to deliver to an AWS Elemental MediaPackage channel that uses MediaPack age v2. For this delivery, you create an HLS output group.	May 17, 2023
Delivery to AWS Elemental MediaPackage v2	The guide includes informati on about setting up an HLS output group to deliver to a AWS Elemental MediaPackage channel that uses MediaPack age v2.	May 17, 2023
Input failover with a AWS Elemental MediaConnect input	The guide now includes information about setting up input failover to work well with a MediaConnect input that is from a flow that implements source redundanc y.	May 12, 2023
Connection fields for downstream system for HLS output groups	There is now informati on about the fields that control reconnecting to the downstream system for an HLS output group.	May 5, 2023
Channel maintenance	The information about managing channel maintenan ce has been expanded.	May 4, 2023

Link UHD Dolby input support	The list of supported audio codecs for Link UHD inputs has been updated to include Dolby Digital and Dolby Digital Plus.	April 18, 2023
Tagging for AWS Elemental Link devices	AWS Elemental Link devices now support AWS resource tagging.	March 27, 2023
MediaLive alerts	The guide now includes a list of the alerts that MediaLive might generate when a channel is running	March 10, 2023
Correction to automatic input failover	We have corrected the description of how MediaLive handles a pipeline failure (scenario 1) in a standard channel. The text and the diagram have been revised.	March 3, 2023
Nielsen watermarks	The section about Nielsen watermarks has been updated to include the new field for specifying a local time zone.	February 20, 2023
Setting up MediaLive as a trusted entity	The section about setting up MediaLive as a trusted entity has been rewritten. The information has been reorganized. But there are no changes to the underlying rules for setting up a trusted entity.	February 14, 2023

Setting up IAM permissions	There is an update to the section about identifying permissions that you must assign to users and other AWS identities. The information has been reorganized. But there are no changes to the guidance, or to the list of services and actions.	February 14, 2023
AWS Identity and Access Management	Updated guide to align with the IAM best practices . For more information, see Security best practices in IAM.	February 14, 2023
Handling of SCTE 35 messages based on description	The chapter includes information that describes how MediaLive handles SCTE 35 messages depending on the ad avail mode and the segmentation descriptor in the message. This information has been corrected to clarify that in splice insert mode, messages with no segmentation descriptor are not treated as ad avails.	February 1, 2023
New metrics	The guide now includes information about the Dropped Frames metric and the SVQ Time metric.	January 26, 2023
Timecode burnin	The user guide now includes information about burning the timecode into output video.	January 20, 2023

Input loss handling	The guide now includes information about the existing input loss behavior feature. You can customize the way that MediaLive handles media when the video input into the channel is lost.	January 13, 2023

Note

- The AWS Media Services are not designed or intended for use with applications or in situations requiring fail-safe performance, such as life safety operations, navigation or communication systems, air traffic control, or life support machines in which the unavailability, interruption or failure of the services could lead to death, personal injury, property damage or environmental damage.
- A component of MediaLive is licensed under the AVC patent portfolio license for the personal and non-commercial use of a consumer to (i) encode video in compliance with the AVC standard ("AVC video") and/or (ii) decode AVC video that was encoded by a consumer engaged in a personal and non-commercial activity and/or was obtained from a video provider licensed to provide AVC video. No license is granted or shall be implied for any other use. A component of MediaLive is licensed under the mpeg-4 patent portfolio license for the personal and non-commercial use of a consumer for (i) encoding video in compliance with the mpeg-4 visual standard (“mpeg-4 video”) and/or (ii) decoding mpeg-4 video that was encoded by a consumer engaged in a personal and non-commercial activity and/or was obtained from a video provider licensed to provide AVC video. No license is granted or shall be implied for any other use. Additional information may be obtained from MPEG-LA, LLC. See http://www.mpegla.com.
- MediaLive may contain Dolby Digital and Dolby Digital Plus, which are protected under international and U.S. copyright laws as unpublished works. Dolby Digital and Dolby Digital Plus are confidential and proprietary to Dolby Laboratories. Their reproduction or disclosure, in whole or in part, or the production of derivative works therefrom without the express permission of Dolby Laboratories is prohibited. © Copyright 2003-2015 Dolby Laboratories. All rights reserved.

AWS Glossary

For the latest AWS terminology, see the AWS glossary in the *AWS Glossary Reference*.