AWS Elemental MediaLive

User Guide

AWS Elemental 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

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) to MediaLive.

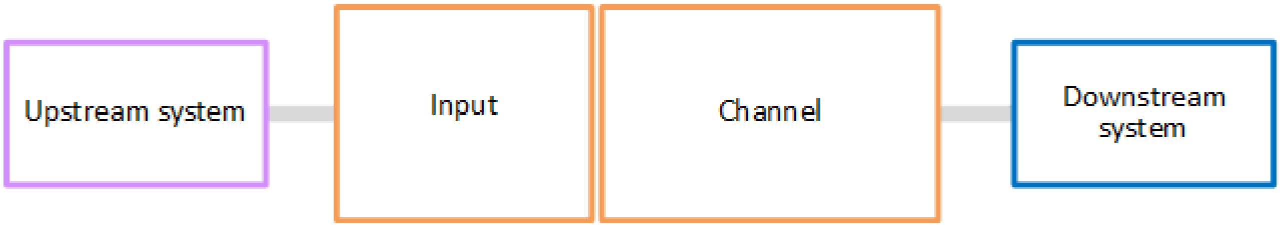
  Examples of an upstream system are a streaming camera or appliance that is directly connected to the internet, or a contribution encoder that is located in a sports stadium where a sports event is being held.

  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.

Pipelines

The processing within MediaLive occurs within one or two pipelines.

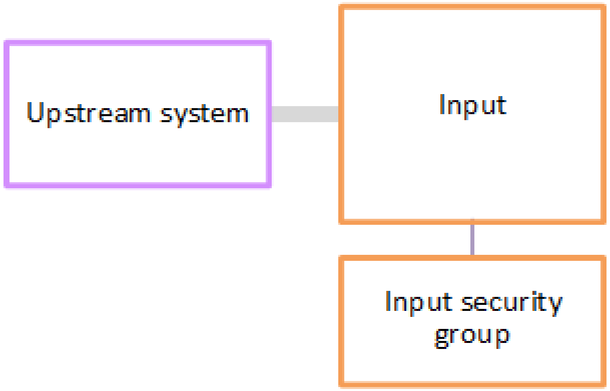
If you set up the workflow so that the channel and inputs have two pipelines (recommended), both pipelines work independently of each other but perform identical processing. Setting up with two pipelines provides resiliency within MediaLive.

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 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

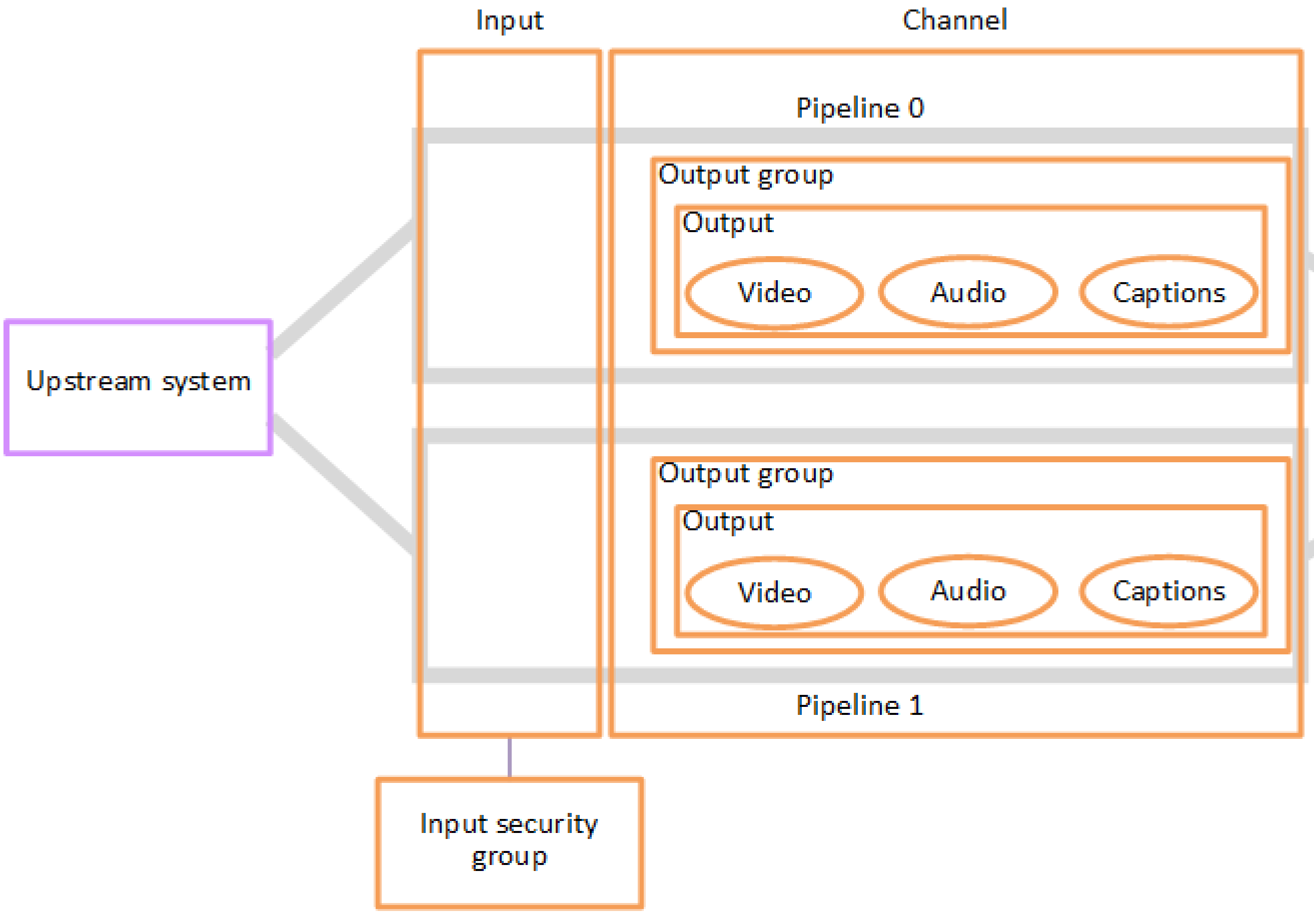
A channel can have several inputs attached to it, but it only ingests source content from one input at a time. (You use the channel *schedule* (p. 3) to set up the channel to switch from one input to another.)

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* (p. 72) and *Setup: Planning the channel* (p. 121).

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 *Resources: MediaLive schedule* (p. 257)

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 AWS Elemental 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 AWS Elemental 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.

Pricing

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

There are two components to pricing: pricing based on the input of the channel that is being processed, and pricing based on the outputs of the channel:

- The input pricing is based on a combination of the input codec, the bitrate of the input, and the resolution of the input. You specify these three characteristics in the input specification when you create the channel. For more information, see the section called “Input specifications settings” (p. 147).
- The output pricing is based on a combination of the output codec, the output frame rate, and the output resolution. You specify these values in the codec, frame rate, width, and height fields in the video settings of each output in the channel. For more information, see the section called “Step 6: Set up video” (p. 204). Note that it is possible to set up the output frame rate to match the frame rate of the input. In this case, the frame rate portion of the pricing calculation uses the rate for "30-60 fps" frame rate; it doesn't use the actual input frame rate.

There are different charges for inputs and outputs when the channel is running compared to when the channel is idle.

As soon as you start a channel, running charges start accruing for inputs and outputs. Running charges continue if you pause one or both pipelines in a channel. Running charges stop accruing only when you stop the channel.

For more information about pricing, see https://aws.amazon.com/medialive/pricing/.

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. MediaLive also includes *constraints* that you can't change. For more information about these constraints, see *Feature rules and limits* (p. 9).

> **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 schedule actions limit is documented in *Feature rules and limits* (p. 9).

The Service Quotas console provides information about MediaLive quotas. Use the Service Quotas console to view default quotas and request quota increases for AWS Elemental MediaLive.

The following table describes the quotas for MediaLive.

Resource or operation	Default quota	Comments
Channels	5	The maximum number of channels that you ca account in the current Region.
Channels with HEVC outputs	5	The maximum number of HEVC channels (char one or more HEVC outputs) that you can create the current Region.
Channels with UHD outputs	1	The maximum number of UHD channels (chann a UHD output) that you can create in this accou Region. For information about the maximum number c these channels, see *Feature rules and limits* (p.
Channels with CDI inputs	2	The maximum number of CDI channels (chann one or more CDI inputs) that you can create in current Region.
Inputs of type *push* (not including VPC push inputs)	5	The maximum number of push inputs (not incl inputs) that you can create in this account in th
Inputs of type *pull*	100	The maximum number of pull inputs that you account in the current Region.
Inputs of type *VPC push*	50	The maximum number of VPC push inputs that this account in the current Region.
Inputs of type *Elemental Link*	100	The maximum number of Elemental Link input in this account in the current Region.
Input security groups	5	The maximum number of input security groups in this account in the current Region.
Multiplexes	2	The maximum number of channels that you ca current Region in this account in the current Re
Reservations	50	The maximum number of reservations that you account in the current Region.

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* (p. 8).

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” (p. 384).
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.
Input number, Elemental Link inputs per MediaLive 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 (p. 351). You can't set up pull inputs as a failover pair.

Resource or feature	Constraint or rule
	The failover pair uses up your maximum number of push inputs.
Input types – for dynamic inputs	Only MP4 file inputs that are stored in Amazon S3 or AWS Elemental MediaStore be set up as dynamic inputs.
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 inputs” (p. 532).
	For the inputs that you attach in order to implement input switching, there are restrictions 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.
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 (p. 374). 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.

Resource or feature	Constraint or rule
	For information about output types, see the section called “Supported containers and downstream systems” (p. 535).
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 (p. 9) in an HLS output group both count towards this limit. • Maximum of three frame capture outputs in each HLS output group. For information about output types, see the section called “Supported containers and downstream systems” (p. 535).
Output video encodes, UHD resolution	A channel that includes an encode with UHD resolution can include only one UHD encode, and it can't include other types of outputs. The maximum number of *channels* with UHD is a quota that you can change, as described in *Quotas* (p. 8). The maximum number of *outputs* in this type of channel 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 for outputs” (p. 537).
	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 resolutions, see the section called “Supported codecs for outputs” (p. 537).
Output – audio encodes	Maximum of 33 audio encodes in one channel.

Limits for other features

Resource or feature	Constraint or rule
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.

Resource or feature	Constraint or rule
	For information about image overlay, see the section called “Image overlays” (p. 392).
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” (p. 432).
Multiplexes	Each multiplex produces only one MPTS. For information about multiplex, see the section called “Multiplex and MPTS” (p. 434).
	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.
	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 (p. 351)	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 (p. 445)	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).

Resource or feature	Constraint or rule
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.
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” (p. 402).
	You can switch to a specific input as many times as you want.
Frequency of API requests	5 steady-state TPS (transactions per second)
	30 burst TPS

Limits for other features

Resource or feature	Constraint or rule
Frequency of API requests	5 steady-state TPS (transactions per second)
	30 burst TPS

Setting up: IAM permissions for AWS Elemental MediaLive

This chapter provides procedures for setting up users to work with AWS Elemental MediaLive. It describes how to grant permissions that are appropriate for the period when you are experimenting with MediaLive, before you start using MediaLive in a production environment.

This chapter covers the following tasks:

- Setting up one or more administrators for the service
- Creating or modifying user identities that have permissions to access AWS Elemental MediaLive and ancillary services that MediaLive typically works with
- Setting up MediaLive as a trusted service

After you perform the procedures in this chapter, you and other users will have permissions that let you successfully follow the *Getting started with AWS Elemental MediaLive* (p. 63).

Important
This chapter includes steps that grant broad permissions to AWS Elemental MediaLive and other services. These permissions are known as AWS Identity and Access Management (IAM) permissions. The permissions are intended to allow you and others in your organization to get started with MediaLive as quickly as possible. These permissions are not suitable for assigning to a wide group of users or for users working in a production environment.
To set up users for production use of AWS Elemental MediaLive, see *Setting up: IAM permissions for production* (p. 21).

Topics

Signing up for AWS Elemental MediaLive

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.

Creating an administrator IAM user

The procedures in this section show how to create an IAM user that has full read/write administrator permissions. This administrator might be you or another person. You set up an administrator by creating a group, and then creating a user that belongs to that group:

- If your organization is new to AWS, follow both steps in this procedure: create the group, and then create the users for that group.
- If your organization is not new to AWS, then the group probably has already been created. Follow only the second step to create users for that group.

To create a full-access administrator group

1. Use your AWS account email address and password to sign in to the AWS Management Console as the AWS account root user.
2. Open the IAM console at https://console.aws.amazon.com/iam/.
3. In the navigation pane, choose **Groups**, and then choose **Create New Group**.
4. On the **Set Group Name** page, for **Group Name**, enter a name such as **`Administrators`**. Choose **Next Step**.
5. On the **Attach Policy** page, choose **Filter: Policy Type**, and then choose **Job function**.
6. In the policy list, select the check box for **AdministratorAccess**, and then choose **Next Step**.
7. On the **Review** page, review the information, and then choose **Create Group**.

Now that you have an administrator group, you are ready to create an IAM user and add the user to your group.

To add an IAM user to the full-access administrator group

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Users**, and then choose **Add user**.
3. On the **Add User** page, for **User name**, enter a name such as **`Administrator`** or **`Admin_2`** (if **`Administrator`** has already been created).
4. For **Access type**, select **AWS Management Console access**.

   For **Console password**, choose **Custom password**, and then enter a password.

   If this administrator is not you, we recommend that you select **Require password reset**.
5. Choose **Next: Permissions**.
6. On the **Set permissions** page, choose **Add user to group**.
7. Select the check box for the group that you created in the preceding procedure, and then choose **Next: Review**.
8. Review the information, and then choose **Create user**. To return to the navigation pane, choose **Close**.

After you create this IAM user with administrator permissions, sign out and sign in again using the administrator credentials.

We highly recommend that from this point forward you always sign in using the IAM administrator credentials instead of your root user credentials, unless AWS requires you to use your root user credentials to perform certain operations. For more information, see AWS Tasks That Require AWS Account Root User Credentials.

Repeat the procedure to set up more administrators (as backups), if needed. Or anyone who is now set up as a full-access administrator can set up more administrators.

Creating a non-administrator IAM user

This section shows how to create non-administrator IAM users and grant those users the following permissions:

- Full read/write access to the following AWS services and features:
  - AWS Elemental MediaLive
  - AWS Elemental MediaConnect
  - AWS Elemental MediaPackage
  - Amazon CloudWatch
  - Amazon CloudWatch Events
  - Amazon CloudWatch Logs
  - Amazon EC2
  - AWS Systems Manager
  - AWS Resource Groups
  - Amazon SNS
  - Amazon VPC
- Limited access to AWS IAM. Users of AWS Elemental MediaLive need some access to IAM in order to use the MediaLive console to set up MediaLive as a trusted entity. This setup is always required when using MediaLive. For more information, see the section called “Setting up as a trusted service” (p. 20).

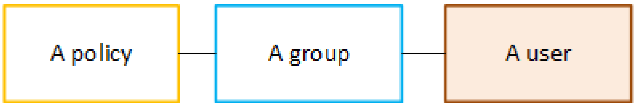
> **Warning**
> These permissions are broad. You should set up only a few users with these permissions and only for the pre-production period of using MediaLive. For information about setting up users for standard production use, see *Setting up: IAM permissions for production* (p. 21).

To set up an IAM user, you follow three main steps:

- Create customer managed policies.
- Create a group and attach the policies to the group.
- Create users and add the users to the group.

Policies grant permissions. Policies are attached to a group. Users belong to a group. Therefore, the users have the permissions of the policies that are attached to the group.

The following diagram shows this relationship.



Topics

Step 1: Create customer managed policies

The procedures in this section show how to create three IAM customer managed policies. A customer managed policy is one that you create and manage. (IAM also includes AWS managed policies, which you can't change.)

Anyone with IAM administrator-level credentials can perform the procedures.

The first procedure shows how to create a policy called `MediaLivePowerAccess` that gives full read/write access to AWS Elemental MediaLive.

The second procedure shows how to create a policy called `MediaConnectPowerAccess` that gives full read/write access to MediaConnect.

The third procedure shows how to create a policy called `MediaLiveTrustedEntityAccess` that gives access to six operations in AWS IAM. These actions allow IAM users to create and update a trusted entity role for AWS Elemental MediaLive by setting the fields in the **IAM role** section on the **Channel and input details** page on the MediaLive console.

To create the MediaLivePowerAccess policy

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Policies**, and then choose **Create policy**. On the **Visual editor tab**, follow the prompts to create a policy with these options:
   - **Service**: **MediaLive**
   - **Actions**: **All MediaLive actions (medialive.*)**
   - **Resources**: Choose **Resources** to open the section, and choose **All resources**.
   - **Request conditions**: Omit this option
3. Choose **Review policy**.
4. On the **Create policy** page, for **Name**, enter `MediaLivePowerAccess`.
5. For **Description**, optionally describe the purpose of this policy. This helps you identify the policy on the dashboard.
6. Choose **Create policy**.

To create the MediaConnectPowerAccess policy

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Policies**, and then choose **Create policy**. On the **Visual editor tab**, follow the prompts to create a policy with these options:
   - **Service**: **MediaConnect**
   - **Actions**: **All MediaConnect actions (mediaconnect.*)**
   - **Resources**: Choose **Resources** to open the section, and choose **All resources**.
   - **Request conditions**: Omit this option
3. Choose **Review policy**.
4. On the **Create policy** page, for **Name**, enter `MediaConnectPowerAccess`.
5. For **Description**, optionally describe the purpose of this policy. This helps you identify the policy on the dashboard.
6. Choose **Create policy**.

To create the MediaLiveTrustedEntityAccess policy

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Policies**, and then choose **Create policy**. On the **Visual editor tab**, follow the prompts to create a policy with these options:
   - **Service**: **IAM**
   - **Actions**: In the filter box under **Specify the actions allowed in IAM**, search for and then select each of these actions:
     - **ListRoles**
     - **GetRolePolicy**
     - **CreateRole**
     - **PassRole**
     - **AttachRolePolicy**
     - **PutRolePolicy**
   - **Resources**: Choose **Resources** to open the section, and choose **All resources**.
   - **Request conditions**: Omit this option
3. Choose **Review policy**.
4. On the **Create policy** page, for **Name**, enter **`MediaLiveTrustedEntityAccess`**.
5. For **Description**, optionally describe the purpose of this policy. This helps you identify the policy on the dashboard.
6. Choose **Create policy**.

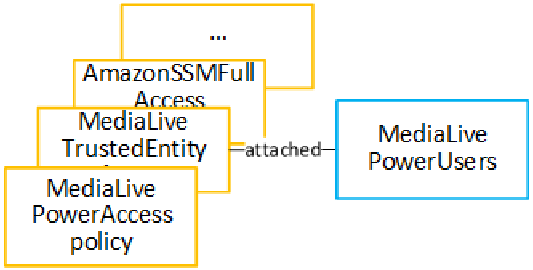
Step 2: Create an IAM group

The procedure in this section shows how to create an IAM group and attach policies. Anyone with IAM administrator-level credentials can perform the procedure. Perform this procedure once, at initial setup. Before you start the procedure, you should have already created the two policies in Step 1: Create Customer Managed Policies (p. 17).

To create a group

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Groups**, and then choose **Create New Group**.
3. On the **Set Group Name** page, for **Group Name**, enter **`MediaLivePowerUsers`**, and then choose **Next Step**.
4. On the **Attach Policy** page, select the check boxes for the following policies:
   - **MediaLivePowerAccess** (customer managed policy)
   - **MediaConnectPowerAccess** (customer managed policy)
   - **MediaLiveTrustedEntityAccess** (customer managed policy)
   - **CloudWatchReadOnlyAccess** (AWS managed policy)
   - **CloudWatchEventsFullAccess** (AWS managed policy)
   - **AmazonEC2FullAccess** (AWS managed policy for access to AWS Virtual Private Network)
   - **AWSElementalMediaPackageFullAccess** (AWS managed policy)
   - **ResourceGroupsandTagEditorFullAccess** (AWS managed policy)
   - **AmazonSSMFullAccess** (AWS managed policy for access to AWS Systems Manager)
   - **AmazonSNSFullAccess** (AWS managed policy)
5. Choose **Next Step**, review your information, and then choose **Create Group**.

This diagram shows how the policies and group are associated.



Step 3: Create or add an IAM user to your group

The procedure in this section shows how to create or edit an IAM user identity. Anyone with IAM administrator-level credentials can perform the procedure. Perform this step for each user.

> **Note**
> This procedure shows how to set up an IAM user for console access, but not for AWS CLI or AWS SDK access. To set up for programmatic access, see the *IAM User Guide*.

Creating an iam user and adding the user to your group

Typically, you create an IAM user identity for an AWS user only if a person doesn't have an existing identity. If the person already has an IAM user identity, you can modify their access (p. 20) instead.

To create an IAM user and add the user to your group

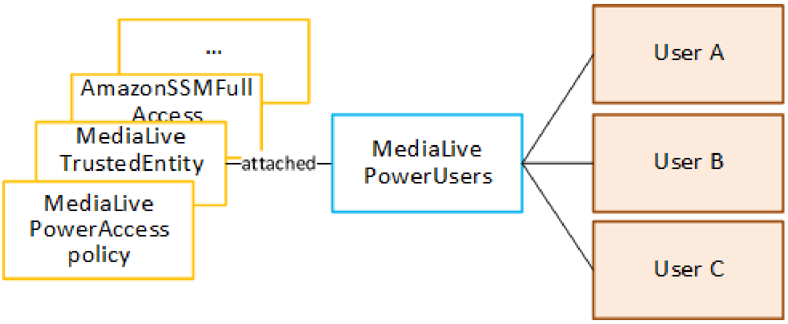
1. Sign in to the AWS Management Console as an administrator, and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Users**, and then choose **Add user**.
3. On the **Add User** page, for **User name**, enter a name for the user.

   For **Access type**, select **AWS Management Console access**.

   For **Console password**, choose **Custom password**, and then enter a password.

   For **Require password reset**, we recommend that you select the check box.
4. Choose **Next: Permissions**.
5. On the **Set permissions for user** page, choose **Add user to group**.
6. Select the check box for the **MediaLivePowerUsers** group that you created in Step 2: Create a Group (p. 18), and then choose **Next: Review**.
7. Choose **Create user**.
8. Optionally choose **Send email** to send an email to this user. Your local email client opens with a draft email that includes the user name and sign-in URL.
9. Choose **Close** to return to the navigation pane.
10. Provide the user with their password (it is not included in the generated email). You must provide the password in a way that complies with your organization's security guidelines.

Repeat the steps to add more IAM users. As an example, the following diagram shows three IAM users that are associated with the same group, **MediaLivePowerUsers**.



Adding an existing IAM user to your group

You can add an existing IAM user to a group that you create for AWS Elemental MediaLive, even if the user is already a member of other groups. In this procedure, you add the user to the **MediaLivePowerUsers** group that you created in Step 2: Create a Group (p. 18).

For more information about IAM users and groups, see *IAM User Guide*.

To add an existing IAM user to your group

1. Sign in to the AWS Management Console as an administrator, and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Users**.
3. In the list of users, choose the user name (don't choose the check box).
4. On the **Summary** page, choose the **Groups** tab. On the **Groups** tab, choose **Add user to groups**, and then select the **MediaLivePowerUsers** group that you created in Step 2: Create a Group (p. 18).
5. Choose **Add to Groups**.

You now have a setup where an IAM user belongs to more than one group: the original groups and the group that you added. If one of the groups has a policy that gives specific permissions to a given service or resource, and another group has a policy that gives different permissions, the policy with the *least permission* applies. One situation in which this rule might apply is if the existing user currently has permissions in IAM that are broader than those in the **MediaLiveTrustedEntityAccess** policy that you created.

Setting up AWS Elemental MediaLive as a trusted service

Every time a user creates a channel, they must attach an IAM role that sets up MediaLive as a trusted entity for that channel. You must give the user the permissions to set up this trusted entity.

You give this permission when you create the user. You create a policy called **`MediaLiveTrustedEntityAccess`** and attach it to the group that the users belong to. For detailed information, see the section called “Step 1: Create customer managed policies” (p. 17).

Setting up: IAM permissions for AWS Elemental MediaLive for a production environment

This chapter provides procedures for setting up users and other AWS identities so that they can use AWS Elemental MediaLive in a production environment. It describes options for imposing restricted controls on users, so that you can set up permissions that conform with the security policies and procedures of your organization.

Before you follow these procedures, do the initial setup described in *Setting up: IAM permissions* (p. 14). Those instructions show you how to grant broad permissions to users for non-production environments. Then return to this chapter to create limited permissions for a production environment.

> **Note**
> For part of the setup described in this chapter, you use the AWS Identity and Access Management (AWS IAM) service to create user and administrator identities. There might be features of IAM, such as cross-account access, that are not covered in this chapter but are appropriate and useful to your deployment. For information about all IAM features, see the AWS IAM User Guide.

In this chapter, we assume the following:

- You are now moving from experimenting with MediaLive to using MediaLive in a production environment.
- You have followed the procedures in *Setting up: IAM permissions* (p. 14) to sign up for MediaLive and to create a full-access administrator user.
- You have followed the procedures in the section called “Creating a non-administrator IAM user” (p. 16) and are therefore familiar with the process for creating IAM users and IAM groups using the IAM console.

This chapter also describes the AWS services that integrate with or depend on MediaLive. For some of these services, you must grant permissions so that users can access the services and use them with MediaLive. For other services, you don't need to grant permissions because the services are fully integrated with MediaLive. Following is a list of the AWS services that are covered in this chapter:

- AWS CloudTrail
- Amazon CloudWatch
- Amazon CloudWatch Events
- Amazon CloudWatch Logs
- Amazon Elastic Compute Cloud (Amazon EC2)
- AWS IAM
- AWS Elemental MediaConnect
- AWS Elemental MediaPackage
- AWS Elemental MediaStore
- AWS Resource Groups (Resource Group Tagging)
- Amazon Simple Notification Service (Amazon SNS)

- Amazon Simple Storage Service (Amazon S3)
- AWS Systems Manager

The following sections describe how to grant permissions to MediaLive and, if needed, the other AWS services.

Topics

Setting up administrators and users

You must set up each person who will use AWS Elemental MediaLive as a IAM user. It is useful to split user identities into three general groups:

- Full-access administrator users. These users have full read/write access to all AWS services, users, and resources, including broad permissions in IAM.

  You already created this user when you followed the procedure in the section called “Creating an administrator IAM user” (p. 14).
- Administrators with limited access. Typically, these users have more permissions than a non-administrator user, but they don't have broad permissions in IAM.

  See the section called “Creating an administrator user with limited access” (p. 22).
- Non-administrator users or "regular users." Typically, these users have broad permissions to MediaLive and to some of the services, such as MediaConnect, that MediaLive interacts with. These users have very limited permissions in IAM.

  See the section called “Creating a non-administrator user ” (p. 25).

We recommend that you set up most users as non-administrator users. Set up only highly trusted users as administrator users.

Creating an administrator user with limited access

If you are a full-access administrator, you can create other administrator users and assign each one a different level of access. These administrator users have more access than non-administrator users ("regular" users), but they have less access than full-access administrator users. They can use AWS Elemental MediaLive in the same way as regular users, but they can also create non-administrative users and set up some of the services that MediaLive integrates with.

For example, you might create an administrator user with the following access:

- For MediaLive and services that integrate with MediaLive, the administrator has the same access as regular users.
- For services that require some setup to work with MediaLive, the administrator has more access than regular users.

- For IAM, the administrator has more access than regular users, but less than full-access administrators.

The following procedure shows how to create an administrative user who has limited access. You start by creating a custom policy with a name such as `MediaLiveAdminAccess`, creating a group called **`MediaLiveAdministrators`**, and attaching the policy to the group. Next, you create the administrator user and add the user to the group. The procedure assumes that the new administrator user does not need permissions to troubleshoot issues with MediaLive other than access issues.

To create a custom policy for a MediaLive administrator

1. Sign in to the AWS Management Console as a full-access administrator, and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Policies**, and then choose **Create policy**. On the **Create policy** page, choose the **Visual editor** tab. This tab is a policy generator that lets you build a policy by selecting actions from a list to add them to the policy.
3. Read the table at the end of this procedure, and create a policy that gives access to the actions that aren't already covered by an existing policy. You don't need to create a policy when we suggest using an existing policy. For information about the purpose of these actions, see the section called “Step 1: Requirements for permissions” (p. 26).
4. To create the policy, follow the prompts on the console. Here are some tips for creating the policy:

   - You can create one policy that covers several services. You don't need to create a policy for each separate service. To create a policy for several services, choose the actions for one service, and then choose **Add additional permissions** at the bottom of the page to set up another service. You might need to move both of the vertical scroll bars to the bottom to display this link.
   - If you do choose to create one policy that covers several services, you might choose to create the policy with actions for one service, save it, then edit the policy to add permissions for another service, and so on.
   - You can choose the **Import managed policy** button to import an existing policy into this policy. The policy actions are copied over (the policy is *not* copied by reference), so after importing you can add and remove actions if you want.

   For full instructions on creating a custom policy, see the *IAM User Guide*.

   The following table shows which actions to include in the policy in order to grant the identified access to the user.

Feature	Corresponding service in IAM	Type of access	Actions to include in the policy
MediaLive Features	**MediaLive**	Full access to MediaLive. It is a good idea for the administrator to be able to work with all MediaLive features.	Use the customer managed policy **MediaLivePowerAccess**. If you followed the procedures in *Setting up: IAM permissions* (p. 14), you created this policy in the the section called “Step 1: Create customer managed policies” (p. 17) section.

Feature	Corresponding service in IAM	Type of access	Actions to include in the policy
Monitoring Channel Health	**CloudWatch**	Limited access to CloudWatch (the same access as non-administrator users).	See the section called "Reference: summary of user access" (p. 50).
Setting Up for Email Notification	**CloudWatch Events**	Full access to CloudWatch Events, to set up users for email notification. (To set up for email notification, users also need access to SNS. See later in this table.)	Use the managed policy **CloudWatchEventsFullAccess**. The administrator might not need all these actions, but giving full access is probably low risk.
Setting Up Channel Logging	**CloudWatch Logs**	Limited access to CloudWatch Logs (the same access as non-administrator users).	See the section called "Reference: summary of user access" (p. 50).
Creating a VPC Input Setting up for delivery to your VPC	**EC2**	Limited access to Amazon EC2 (the same access as non-administrator users)	See the section called "Reference: summary of user access" (p. 50).
Setting Up User Identities for MediaLive	**IAM**	Limited access to manage users, groups, policies, and trusted entity roles.	The action `ChangePassword` And all actions that have any of these strings in their name: "User", "Group", "Policy", "Policies", "Role", "AccessKey", "LoginProfile". Except don't include actions that also have the string "Instance", or the string "ContextKeys"
Setting Up Email Notification	**SNS**	Full access to **SNS**, to set up email notification for users. (To set up for email notification, users also need access to CloudWatch Events. See earlier in this table.)	Use the managed policy **AmazonSNSFullAccess**. The administrator might not need all these actions, but giving full access is probably low risk.

Feature	Corresponding service in IAM	Type of access	Actions to include in the policy

Creating a Password Parameter in AWS Systems Manager Parameter Store	**Systems Manager**	Full access to actions relating to AWS Systems Manager Parameter Store. This access allows the administrator to create and delete password parameters.	`*Parameter`
			`*Parameters`
			`*ParameterHistory`
			`GetParametersByPath`

To create a group for your custom policy or policies

1. If necessary, sign in to the AWS Management Console as a full-access administrator, and open the IAM console at https://console.aws.amazon.com/iam/. (You might still be signed in.)
2. In the navigation pane, choose **Groups**. Follow the prompts to create a group with a name such as **`MediaLiveAdministrators`**.
3. Attach the policy or policies that apply to this administrator.

For full instructions on creating a group and attaching a policy, see the *IAM User Guide*.

To create an administrator user and add the user to your group

1. If necessary, sign in to the AWS Management Console as a full-access administrator, and open the IAM console at https://console.aws.amazon.com/iam/. (You might still be signed in.)
2. In the navigation pane, choose **Users**. Follow the prompts to create a user using the name of the person who will be the administrator.
3. In the step to set permissions for the user, choose **Add user to group**, and then select the group that you created.
4. Follow the prompts to finish creating the user.

For full instructions for creating an administrator, see the *IAM User Guide*.

Creating a non-administrator user

This section describes how to create non-administrator users ("regular users") by using IAM to create groups, attach policies to each group, and add the users to the group.

If you are new to AWS or if you have been using AWS for only a few weeks, we recommend that you read this entire section.

If you have more experience using other AWS services, read the section called “Step 1: Requirements for permissions” (p. 26). Then, based on your decisions, modify or create policies and groups in the usual way.

Summary of Steps

To create IAM users with access to AWS Elemental MediaLive, you must perform several steps:

- Identify the permissions that users need for MediaLive and other services.

- Identify the different sets of users that you need. Each set will become an IAM group.
- Identify the managed and custom policies that will provide the access required for the sets of users. Create the custom policies.
- Create the groups that you have identified, and attach the managed and custom policies.
- Create each user and add them to the appropriate group.

Topics

Step 1: Identify requirements for permissions for users

You must identify the IAM permissions that you need to grant to users, for AWS Elemental MediaLive features and for ancillary services that MediaLive always interacts with.

To do that, you should understand the MediaLive workflows for your organization and the different AWS services that the workflows use.

You might not want all regular users to have the same permissions. For example, you might be able to group regular users into three sets: users who can start channels and watch channel activity, users who have some write capabilities, and advanced users who can do everything. As you identify these permissions, think about how many different sets of users you need.

Topics

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 *Resources: MediaLive reservations* (p. 251).

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

View alerts for running channels	MediaLive	`ListAlerts`

Permissions	Service name in IAM	Actions
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:ListInputs",` directly in the JSON. You can perform all these steps in the IAM console.		

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

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`
Change the class for a channel	MediaLive	`UpdateChannelClass`
Run channels	MediaLive	`StartChannel` `StopChannel`
Pause channels	MediaLive	Pause is part of the schedule feature, above.

Permissions	Service name in IAM	Actions
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	CloudFront	`ListDistributions\`
		`DescribeDistribution`
		`CreateDistribution`

Permissions	Service name in IAM	Actions
MediaPackage as an output destination. Use the workflow wizard to delete a workflow that includes a CloudFront distribution.		`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 “Monitoring using CloudWatch events” (p. 327).)

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 "Monitoring using CloudWatch Logs" (p. 330).

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` `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" (p. 220).

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” (p. 503).

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 AWS Identity and Access Management—trusted entity role

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 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 containers in AWS Elemental MediaStore so that a specified channel can access some containers and not others.

Every time a user creates a channel, they must attach an IAM role that sets up MediaLive as a trusted entity for that channel. The user makes this attachment using the **IAM role** pane on the **Create channel** page on the MediaLive console.

You must decide what access you need to give to users for working in this **IAM role** pane.

If you followed the procedures in the section called “Creating a non-administrator IAM user” (p. 16) to set up users for the period when you are experimenting with MediaLive, then you already set up this trusted entity role. You set it up by creating the `MediaLiveAccessRole` role. However, you should still read this section to determine if `MediaLiveAccessRole` is suitable for your organization when you are working in a production environment.

Topics

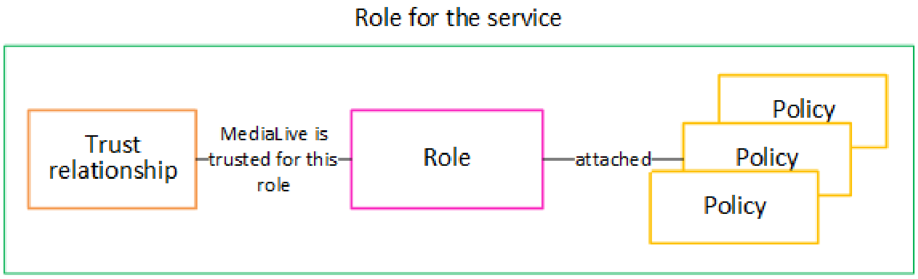
About the trusted entity role

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 AWS Elemental MediaStore 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 containers in MediaStore.

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:

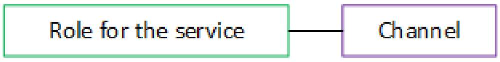
"MediaLive is allowed to assume this role in order to perform the operations on the resources that are specified in the policies."



After this role is created, the role must be attached to a specified 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."

Creating this attachment at the channel level allows each channel to give MediaLive access to different operations and, especially, different resources.



Options for implementing the role

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

Simple option

The simple option typically applies when users in your organization are using AWS Elemental MediaLive to encode the organization's own assets (not assets belonging to customers), and you don'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 can attach that role to the channels that they work with.

The simple option works only on the MediaLive console. It can't be performed using the AWS CLI, for example.

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, see the section called "Requirements for permissions for the simple option" (p. 34).

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. Therefore, you need to create several access roles, each of which 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.

If the complex option is applicable to your deployment, see the section called "Setting up AWS Elemental MediaLive as a trusted service" (p. 44).

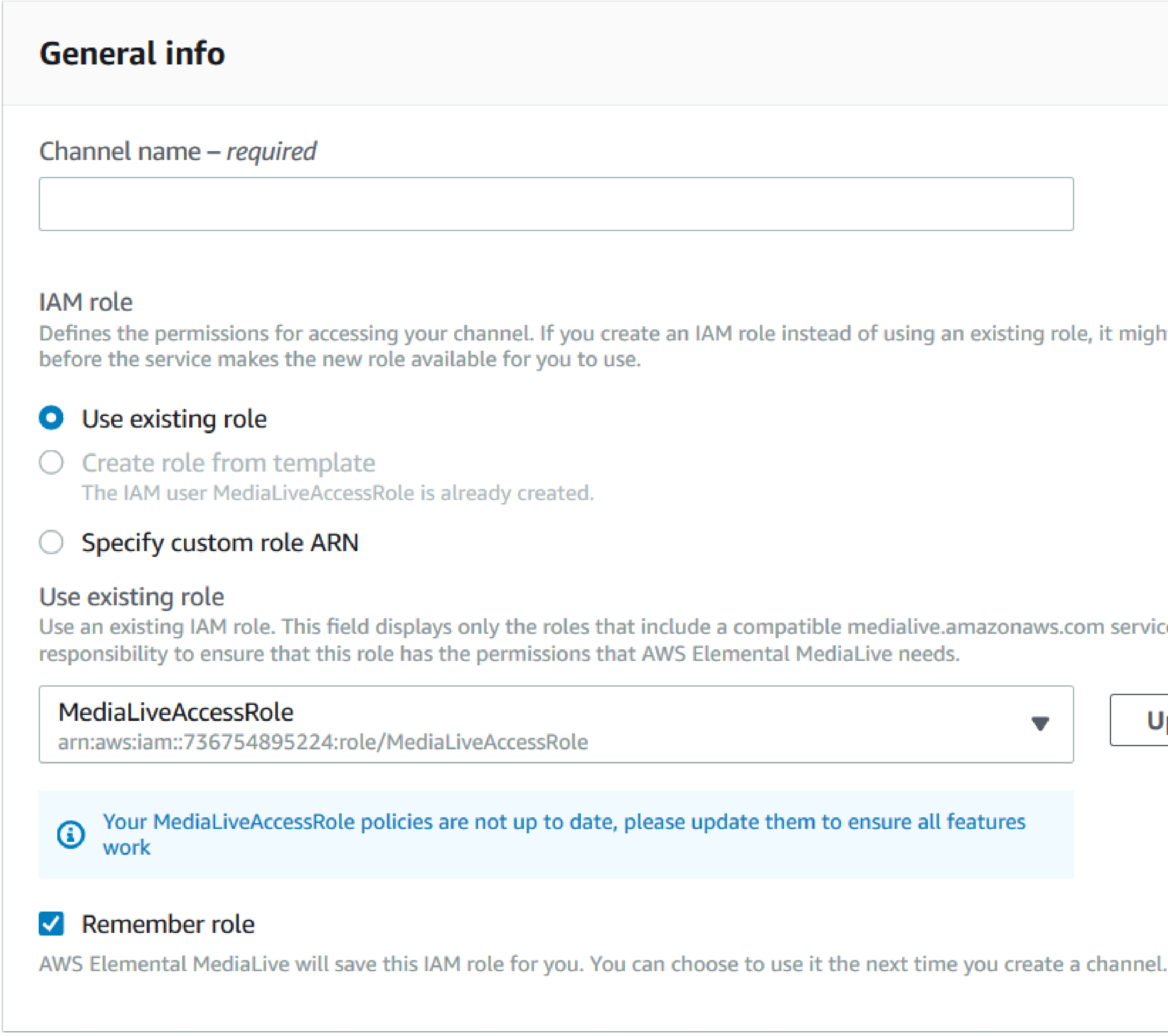
Requirements for permissions for the simple option

Read this section if you decide that the simple option (p. 33) for the trusted entity is appropriate to your deployment.

(To set up for the complex option, see the section called "Setting up AWS Elemental MediaLive as a trusted service" (p. 44).)

For users to work in the **IAM Role** section on the **Channel and input details** pane, they must have access to specific IAM actions.

The following screenshot shows the **IAM Role** section on the **Channel and input details** pane as it appears when you start to create a channel.



General info

Channel name – *required*

IAM role

Defines the permissions for accessing your channel. If you create an IAM role instead of using an existing role, it migh
before the service makes the new role available for you to use.

● Use existing role

○ Create role from template

The IAM user MediaLiveAccessRole is already created.

○ Specify custom role ARN

Use existing role

Use an existing IAM role. This field displays only the roles that include a compatible medialive.amazonaws.com servic
responsibility to ensure that this role has the permissions that AWS Elemental MediaLive needs.

MediaLiveAccessRole
arn:aws:iam::736754895224:role/MediaLiveAccessRole

Your MediaLiveAccessRole policies are not up to date, please update them to ensure all features work

☑ Remember role

AWS Elemental MediaLive will save this IAM role for you. You can choose to use it the next time you create a channel.

You must set up users as follows:

- Users must be able to choose `MediaLiveAccessRole` from the selection field that accompanies the **Use existing role** field.
- Users must be able to choose 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 all users these permissions.)
- Users do not need to be able to use the **Specify custom role ARN** field. They will use `MediaLiveAccessRole`. They will never use a custom role.
- Users must be able to choose the **Update** button, in order to update the `MediaLiveAccessRole` from time to time.

The following table shows the service and action in IAM that you must grant to regular users with the simple option.

Permissions	Service name in IAM	Actions

Choose the **Create role from template option**	IAM	`CreateRole`
		`PutRolePolicy`
		`AttachRolePolicy`

Choose **MediaLiveAccessRole** from the list in **Use existing role**	IAM	`ListRole` `PassRole`

Choose **Update**	IAM	`GetRolePolicy`
		`PutRolePolicy`
		`AttachRolePolicy`

Requirements for AWS Elemental MediaConnect

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

The user needs permissions to perform actions in MediaConnect when they use the MediaLive workflow wizard. The user doesn'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 includes a source from MediaConnect.
	MediaConnect	`List*`
		`Describe*`
		`Create*`
		`Delete*`

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 (p. 74). (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*`

Permissions	Service name in IAM	Actions

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” (p. 27).
- 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`

Permissions	Service name in IAM	Actions

Modify and Delete	Systems Manager	`DeleteParameter`
		`DeleteParameters`
		`DescribeParameters`
		`GetParameter`
		`GetParameterHistory`
		`GetParameters`
		`GetParametersByPath`

Step 2: Identify categories of users

After you have identified the permissions that your users need, you must identify the different categories of users that you need. You identify different categories based on the different operations they should be allowed to perform.

This requirements analysis must be performed by a person in your organization who understands the AWS Elemental MediaLive workflows for your organization and the different AWS services that the workflows use.

To identify categories of users

1. Refer to the sections in the section called "Step 1: Requirements for permissions" (p. 26), and decide whether all of your users should have the same permissions on all the services or whether some users should have one set of permissions while other users have another set.
2. Group these different categories into, giving each category a name.
3. When looking at operations, keep in mind that you could decide that no regular user should have certain permissions—only an administrator should have those permissions.

For example, perhaps you identify three categories of users:

- Basic users –These users can start and stop channels and view metrics for channels, but have no write permissions.
- Read/write users – These users have nearly full permissions, but they can't create password parameters in AWS Systems Manager Parameter Store.
- Advanced users – These users have full permissions on the services identified in the section called "Step 1: Requirements for permissions" (p. 26). They are nearly as powerful as a restricted administrator, except that they can't set up users.

Step 3: Create the Custom Policies

Any person who is an administrator can perform this procedure. Follow this procedure once, when setting up users for production.

After you identify the different collections of operations that your sets of users require, you must determine which collections have a corresponding managed policy or custom policy that already exists, and which require a new custom policy.

To identify and create custom policies

1. Look at each set of users that you identified, and look at the collections of operations for those users. For each service, determine which collections have a corresponding managed policy and which require a custom policy.

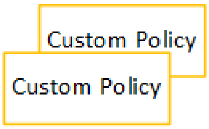
   For example, for CloudWatch Events, there is a managed policy called `CloudWatchEventsFullAccess` that corresponds to "events:*". But there is no policy that contains only the operations required to create a password parameter. You must create a custom policy for that.
2. In IAM, create custom policies as applicable, using the IAM policy generator. This generator lets you choose the service from a list, and then choose operations from a list. As a best practice, give the policy a name that starts with the service name, `medialive`.

   To create the policy, follow the prompts on the console. Here are some tips for creating the policy:

   - You can create one policy that covers several services. You don't need to create a policy for each separate service. To create a policy for several services, choose the actions for one service, and then choose **Add additional permissions** at the bottom of the page to set up another service. (You might need to move both of the vertical scroll bars to the bottom to find **Add additional permissions**.)
   - You can choose **Import managed policy** to import an existing policy into this policy. The policy actions are copied over (the policy is *not* copied by reference), so after importing you can add and remove actions if you want.

   For detailed instructions for creating a policy, see *IAM User Guide*.

The following example assumes that you created two custom policies.



Step 4: Create the groups

Any person who is an administrator can perform this procedure. Follow the procedure once, when setting up users for production.

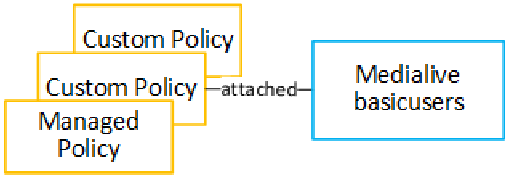
After you identify the different sets of users that your deployment requires, you must create a group for each set.

To create the groups

1. In IAM, choose **Groups**, and then use the **Create New Group Wizard** to create a group for each set of users that you identified in the section called “Step 2: Identify categories of users” (p. 40). See https://docs.aws.amazon.com/IAM/latest/UserGuide/id_groups_create.html and follow the steps for creating groups using the console.

   As a best practice, assign group names that start with the service name, `medialive`.
2. The **Create New Group Wizard** includes a step for attaching policies to the group as you create it. Make sure to attach the managed and custom policies that you have identified.

The following example assumes that you created a group called `medialivebasicusers` and associated the two custom policies plus one managed policy.



Step 5: Create or modify each IAM user

Any person who is an administrator can perform this step. Follow this procedure when setting up users for production. After the initial setup, perform this procedure whenever you need to set up a new user.

After you create the groups and attach the policies to each group, you must create the users and attach each to the appropriate group.

This procedure describes how to set up the user for console access, but not for AWS CLI or AWS SDK access. To set up for programmatic access, see *IAM User Guide*.

Topics

Create a User

Typically, you create a new user identity for an AWS user only if a person does not have an existing identity. If the person already has a user identity, modify their access instead.

To create a user

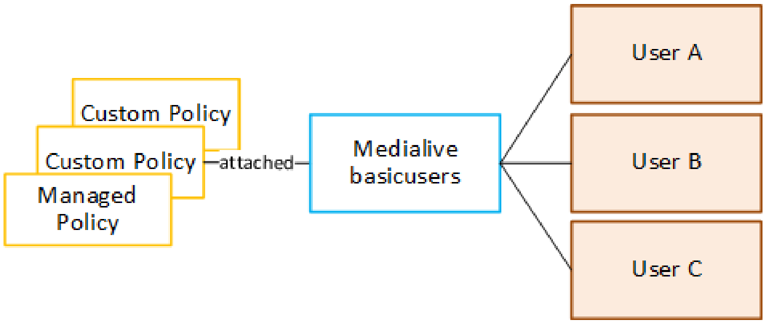
1. Make sure that you know which group (the section called “Step 4: Create the groups” (p. 41)) that you want to add each user to. Make sure that you have already created this group (the section called “Step 4: Create the groups” (p. 41)).
2. Sign in to the AWS Management Console as an administrator, and open the IAM console at https://console.aws.amazon.com/iam/.
3. In the navigation pane, choose **Users**, and then choose **Add user**.
4. On the **Add User** page, for **User name**, enter a name for the user.

   For **Access type**, select **AWS Management Console access**.

   For **Console password**, select **Custom password** and enter a password.

   For **Require password reset**, we recommend that you select the check box.
5. Choose **Next: Permissions**.
6. On the **Set permissions for user** page, choose **Add user to group**.
7. Select the check box for the appropriate group for this user, and then choose **Next: Tags**.
8. Add tags if your organization has a policy to create tags for users. For more information, see the section called “Tagging resources” (p. 479). Then choose **Next: Review**.
9. Choose **Create user**.
10. Optionally, choose **Send email** to send an email to this user. Your local email client opens with a draft email that includes the user name and sign-in URL.
11. Choose **Close** to return to the navigation pane.
12. Provide the user with their password (it is not included in the generated email). You must provide the password in a way that complies with your organization's security guidelines.

The following example assumes that you created three users and associated all of them with the same group, `medialivebasicusers`.



Modify an existing user

If a user identity already exists for someone who will use AWS Elemental MediaLive, you can set them up for use in a production environment by modifying their user identity to make them a member of the relevant group (in addition to the group or groups where they are already members.)

You need to know the group that each user needs to belong to; see the section called “Step 4: Create the groups” (p. 41).

One situation in which it is very useful to modify an existing non-administrator user is if you followed the procedures in *Setting up: IAM permissions* (p. 14) to set up users for the period when you are experimenting with MediaLive. Following that procedure, you created a policy called `MediaLiveAccessUser` and a group called `MediaLivePowerUsers.` You can now take away the broad permissions that you gave those users and "move" the users from the `MediaLivePowerUsers` group to one of the groups that you created in the section called “Step 4: Create the groups” (p. 41). There is no need to delete these users and create them again.

To modify a user

1. Sign in to the AWS Management Console as an administrator, and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Users**.
3. In the list of users, choose the user name (don't select the check box).
4. On the **Summary** page, choose the **Groups** tab:

   - To add this user to another group, on the **Groups** tab, choose **Add user to groups** and follow the prompts to choose the group.
   - To remove this user from a group, choose the **X** icon beside the group name and follow the prompts.

     You might want to remove the user from the `MediaLivePowerUsers` group but keep them in groups that give access to other AWS services.

   You might now have the setup where a user belongs to more than one group, the original groups and the groups that you added. If one of the groups has a policy that gives specific permissions to a given service or resource, and another group has a policy that gives different permissions, the policy with the *least permission* applies.

Step 6: Setting up required data

After you set up users with the appropriate access, you or another administrator should provide users with the information that they need to use MediaLive:

- Provide each user with a list of the MediaLive operations that they have access to. To prevent user frustration, make sure that users know which console pages they can't display. Make sure to include information about the channel metrics that users can't display on the **Channel details** page.
- If some external servers require user credentials, and only one or two users or administrators are responsible for creating password parameters in the AWS Systems Manager Parameter Store, make sure that those users are aware of their responsibility.

  Also make sure that those users provide other users with the password parameters that those other users need.
- If a user or administrator is responsible for setting up other users for email notification, let that user know. Or if each user is responsible for setting up their own email notification, let each user know. Users can read the section called "Monitoring using CloudWatch events" (p. 327) for instructions.
- For the MediaLive role, if you chose the simple option (p. 33), make sure users know that the only role they will ever choose is the `MediaLiveAccessRole`.

  If you set up for the complex option (p. 34), let the user or administrator who must create roles know that they must give other users a list of the roles (the list of role ARNs) that each user can use.

Setting up AWS Elemental MediaLive as a trusted service

You need to read this section only if you determined in the section called "Options for implementing the role" (p. 33) that the simple option for setting up the trusted entity role does not work for your deployment.

This section describes how to implement the complex option. It provides the following information:

- Background information about how the trusted entity role is created and used in the complex option.
- Instructions for identifying the trusted entity role or roles that your deployment needs and creating these roles.
- Instructions for granting limited permissions to regular users so that they can use only specific trusted entity roles.

If you are not familiar with the purpose of the trusted entity role, first read the section called "About the trusted entity role" (p. 33) and the section called "Options for implementing the role" (p. 33).

Topics

How the trusted entity is created and attached

This section applies if you have determined that your deployment should implement the complex option for the trusted entity role, as described in the section called "Options for implementing the role" (p. 33).

With complex options, the process for creating trusted entity roles and attaching a specific role when creating a channel typically works as follows:

- Process for creating a role – An administrator creates the roles using IAM. They don't use the **IAM role** pane on the **Create channel** page on the AWS Elemental MediaLive console. They create these roles as part of the initial deployment.
- Process for attaching a role – After the required roles are created, the administrator gives each regular user a list of the roles and the channels that each role applies to.

  Each user might have a different list of roles; they will have only the roles that apply to the channels that they work with.

  When a user who has permission to create a channel is working on the **Create channel** page, they will display the **Channel and input details** pane. In the **IAM Role** section, the user will choose **Specify custom role ARN** and enter the role name in the field by typing or pasting.

An administrator therefore must perform the following setup:

- Set up all the trusted entity roles that your deployment requires. See the section called "Creating trusted entity roles" (p. 45).
- Set up regular users with restricted permissions for working with roles. You must also make sure that you have not granted certain permissions; granting those permissions would give the regular users permissions that are too broad for the complex option. See the section called "Setting up permissions for non-administrator users" (p. 48).

Creating trusted entity roles

This section applies if you have determined that your deployment requires the complex option for the trusted entity role, as described in the section called "Options for implementing the role" (p. 33).

This section describes how to create a role, policy, and trust relationship, as described and illustrated in the section called "About the trusted entity role" (p. 33).

Topics

Step 1: Determine the access 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 containers in MediaStore so that a specified channel can access some containers and not others.

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.

To determine the access requirements for MediaLive

1. See the table in the section called “Reference: summary of trusted entity access” (p. 56) 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.

   After you perform this analysis, you might determine that you need three different policies for MediaStore, four different policies for Amazon S3, and three policies for AWS Systems Manager Parameter Store.
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.

   For example, for one workflow, you might need policy X for MediaStore, policy A for Amazon S3, and policy 1 for AWS Systems Manager Parameter Store. For the second workflow, you might need policy Y for MediaStore, policy B for Amazon S3, and policy 1 for AWS Systems Manager Parameter Store. For the third workflow, you need the same policies as for the first workflow.
6. Identify the number of roles that you need. You need one role for each unique combination of policies. Following our example, you need two roles: one role for the first and third workflows, and another for the second workflow.
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 2: Create policies

Any person who is an administrator can create a policy.

In the section called “Step 1: Determine requirements” (p. 45), someone in your organization identified the policy or policies that you need to create.

Create those policies now in IAM.

To create a custom policy for the MediaLive trusted entity role

1. If necessary, sign in to the AWS Management Console as a full-access administrator, and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Policies**, and then choose **Create policy**. On the **Create policy** page, choose the **Visual editor** tab. This tab is a policy generator that lets you build a policy by selecting actions from a list to add them to the policy.

   To create the policy, follow the prompts on the console. Here are some tips for creating the policy:

   - You can create one policy that covers several services. You don't need to create a policy for each separate service. To create a policy for several services, choose the actions for one service, and then choose **Add additional permissions** at the bottom of the page to set up another service. (You might need to move both of the vertical scroll bars to the bottom to find **Add additional permissions**.)

- You can choose the **Import managed policy** button to import an existing policy into this policy. The policy actions are copied over (the policy is *not* copied by reference), so after importing you can add and remove actions if you want.

For full instructions on creating a custom policy, see the *IAM User Guide*.

To create the policy, follow the prompts on the console. Here are some tips for creating the policy:

- You can create one policy that covers several services. There is no need to create a policy for each separate service. To create a policy for several services, choose the actions for one service, and then choose the **Add additional permissions** button at the bottom of the screen to set up another service. You may need to move both the vertical scroll bars to the bottom to reveal this button.
- You can choose the **Import managed policy** button to import an existing policy into this policy. The policy actions are copied over (the policy is *not* copied by reference), so after importing you can add and remove actions if you want.

For full instructions on creating a custom policy, see the *IAM User Guide*.

Step 3: 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 “Step 1: Determine requirements” (p. 45), someone in your organization identified the roles that you need to create. Create those roles now using IAM.

To create a role and attach a policy to it

1. Sign into the AWS Management Console as an administrator, and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Roles**.
3. On the **Role** page, choose **Create role**.
4. On the **Create role** page, in the **Select type of trusted entity** section, choose **AWS service** (the default).
5. In **Choose the service that will use this role**, choose **EC2**.

   You choose EC2 because MediaLive is not currently included in this list. Choosing EC2 lets you create a role; in a later step, you will change this role to mention MediaLive instead of EC2.
6. Choose **Next: Permissions**.
7. In the **Attach permissions policies** section, select all the policies that apply for this role, and then choose **Next: Tags**.
8. Add tags if your organization has a policy to create tags for resources. For more information, see the section called “Tagging resources” (p. 479). Then choose **Next: Review**.
9. Choose **Next: Review**.
10. For **Role name**, enter a name. We recommend that you don't use the name `MediaLiveAccessRole` because it is reserved for the simple option (p. 33). Instead, use a name that includes `medialive` and describes this role's purpose.
11. For **Trusted entities**, Amazon EC2 (`ec2.amazonaws.com`) is displayed as the trusted entity, but you will modify that line in the next procedure.

12. Choose **Create role**.

Step 4: Revise the trust relationship

Any person who is an administrator can perform this procedure.

When you created the role and established the trusted relationship, you chose EC2 as the service. You must now modify the role so that the trusted relationship is between your AWS account and MediaLive.

To change the trust relationship to MediaLive

1. On the **Summary** page for the role (which should still be displayed), choose **Trust relationships**.
2. Choose **Edit trust relationship**.
3. For **Edit Trust Relationship**, for **Policy Document**, change `ec2.amazonaws.com` to `medialive.amazonaws.com`.

   The policy document should now look like this:

```
{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Principal": {
"Service": "medialive.amazonaws.com"
},
"Action": "sts:AssumeRole"
}
]
}
```

4. Choose **Update Trust Policy**.
5. On the **Summary** page, make a note of the value in **Role ARN**. It looks like this:

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

   In the example, `111122223333` is your AWS account number.
6. Make a list of all the role ARNs, and include a description of the workflow and users for each. You will need this list in the section called “Step 6: Setting up required data” (p. 44).

Setting up permissions for non-administrator users

This section applies if you have determined that your deployment requires the complex option for the trusted entity role, as described in the section called “Options for implementing the role” (p. 33).

This section describes how you set up permission for regular users to work with the trusted entity roles that you created in the section called “Creating trusted entity roles” (p. 45).

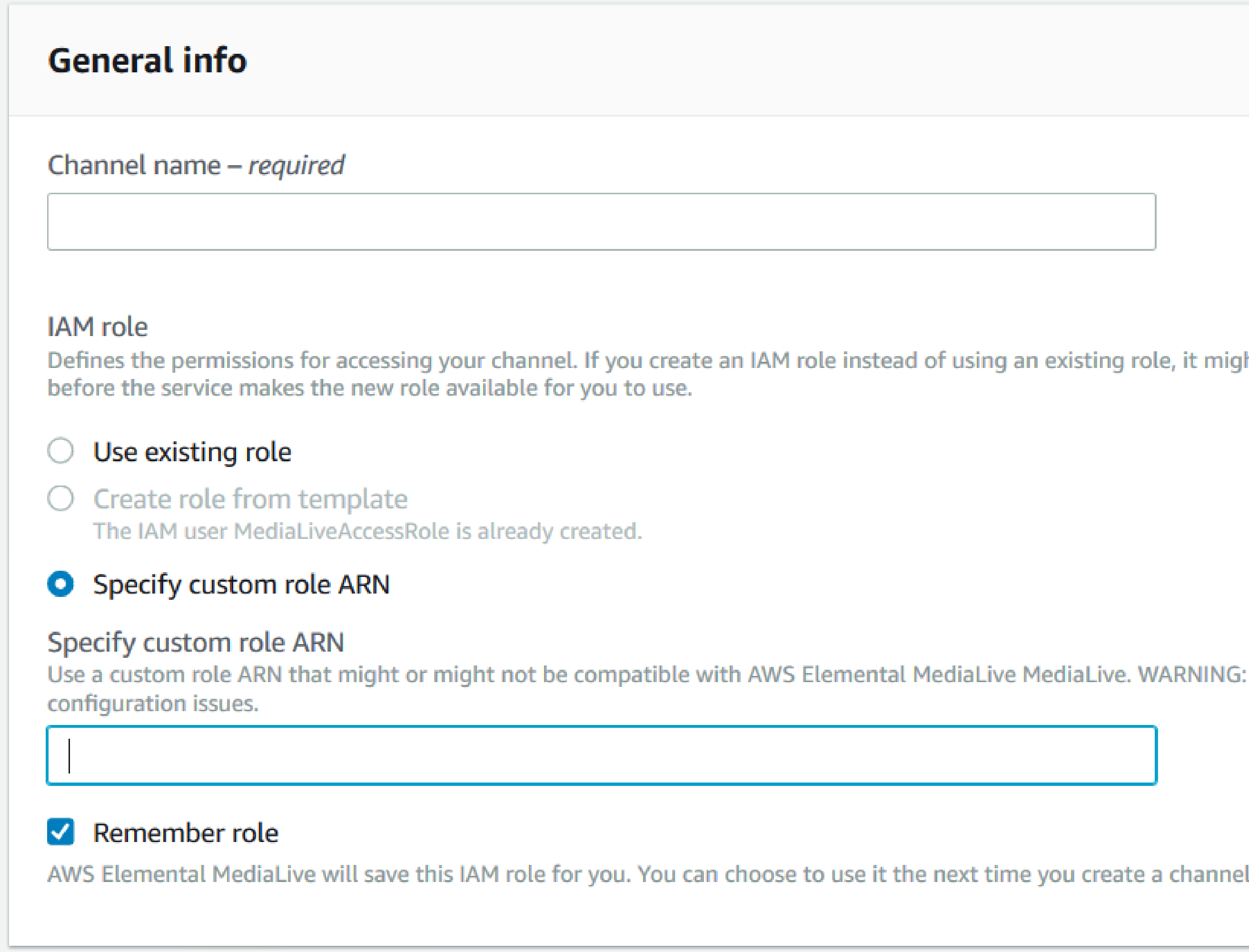
Topics

Identifying permissions

With the complex option, regular users don't create trusted entity roles. But they will attach existing roles to the channels that they create by completing the **IAM Role** section in the **Channel and input details** pane.

To work with this section, regular users therefore must have access to specific IAM actions.

The following screenshot shows the **IAM Role** section on the **Channel and input details** pane as it appears when you start to create a channel.



General info

Channel name – *required*

IAM role

Defines the permissions for accessing your channel. If you create an IAM role instead of using an existing role, it migh
before the service makes the new role available for you to use.

○ Use existing role

○ Create role from template

The IAM user MediaLiveAccessRole is already created.

● Specify custom role ARN

Specify custom role ARN

Use a custom role ARN that might or might not be compatible with AWS Elemental MediaLive MediaLive. WARNING:
configuration issues.

☑ Remember role

AWS Elemental MediaLive will save this IAM role for you. You can choose to use it the next time you create a channel

You must set up permissions for regular users so that they can access only specific fields on the **IAM Role** section in the **Channel and input details** pane. Typically, you must set up as follows:

- Users must *not* be able to choose the selection field that accompanies the **Use existing role** field. You probably want to disable this selection field because you don't want users to choose a role from the list that accompanies this field. If this field is enabled for a user, that user can view all the roles that are created in the account, which would defeat the requirement to restrict access so that users can view and attach only specific roles.
- Users must *not* be able to choose the **Create role from template** field. Regular users do not create roles.

- Users must be able to enter values into the entry field that accompanies the **Specify custom role ARN** field. When this entry field is enabled, the user can enter or paste one of the role names that you provide.
- Users do *not* need to be able to choose the **Update** button because this button only ever appears in implementations that use the `MediaLiveAccessRole`. The complex option does not use this role; therefore, this button never appears.

To ensure that users interact with this section of the console in this restricted way, you must grant access to only one IAM action, as shown in the following table.

Permissions	Service Name in IAM	Actions
Attach	IAM	`iam:PassRole`

Equally important, you must make sure that you do *not* grant access to the following actions:

- `iam:ListRole`
- `iam:CreateRole`
- `iam:PutRolePolicy`
- `iam:AttachRolePolicy`

Setting up permissions

This section applies if you have determined that your deployment requires the complex option for the trusted entity role, as described in the section called “Options for implementing the role” (p. 33).

You must create a policy for the IAM service to set up these regular users with the permissions that they need for the complex option. You must also attach that policy to the group that those regular users belong to.

To set up permissions

1. Follow the steps in the section called “Step 3: Create the Custom Policies” (p. 40) with these differences:
   - Create a policy with a name such as `MediaLiveTrustedEntityRegularUserAccess`.
   - Include only the actions that you identified in the section called “Identifying permissions” (p. 49).
2. In the group that you created or will create for regular users (see the section called “Step 4: Create the groups” (p. 41)), include this policy.
3. Identify any other policies that relate to the IAM service, and detach them from this group.

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. This table is a summary of the tables found in the section called “Step 1: Requirements for permissions” (p. 26). 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.

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`
			`UpdateInput`
			`UpdateInputDevice`
			`UpdateInputSecurityGroup`

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

	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`
	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`

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

	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`

	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`

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	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`

	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 MediaPackage output.
			`List*`
			`Describe*`
			`Create*`
			`Delete*`

	MediaStore	Use the workflow wizard to 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*`

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

Monitoring Channel Health	CloudWatch		`ListMetrics`
			`GetMetricData`
			`GetMetricStatistics`

Setting Up Events	CloudWatch Events		All actions The managed policy `CloudWatchEventsFullAccess` provides these permissions
Setting Up Channel Logging	Amazon CloudWatch Logs	View logs	`FilterLogEvents` `GetLogEvents`
		Set retention policy	`DeleteRetentionPolicy` `PutRetentionPolicy`

Simple Option for the Trusted Entity Role	IAM	Create the `MediaLiveAccessRole`	`CreateRole`
			`PutRolePolicy`
			`AttachRolePolicy`

		Choose the `MediaLiveAccessRole`	`ListRole` `PassRole`

		Update the `MediaLiveAccessRole`	`GetRolePolicy`
			`PutRolePolicy`
			`AttachRolePolicy`

Setting 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`
			`DeleteParameters`
			`DescribeParameters`
			`GetParameter`
			`GetParameterHistory`
			`GetParameters`
			`GetParametersByPath`
			`PutParameter`

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

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 (p. 45).

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	MediaLive doesn't need access for this task. Only the users need access.	

Service	Tasks	Type of access required	Suggested actions or policy
	alerts that are sent to CloudWatch Events.		

CloudWatch Logs	Sending channel log information to CloudWatch Logs when a channel is running.	When the channel is running, MediaLive must be able to send log messages to CloudWatch Logs.	`CreateLogGroup`
			`CreateLogStream`
			`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`
			`CreateNetworkInterfacePermission`
			`DescribeNetworkInterfaces`
			`DescribeSecurityGroups`
			`DescribeSubnets`

	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`
			`DeleteNetworkInterfacePermiss`
			`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`
			`CreateNetworkInterfacePermiss`
			`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`

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.

	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.	

Service	Tasks	Type of access required	Suggested actions or policy
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`

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.	

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`

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” (p. 38).	When the channel is running, MediaLive must have read access to the AWS Systems Manager Parameter Store.	The managed policy `AmazonSSMReadOnlyAccess`

Setting up: IAM permissions for distributors of AWS Elemental Link

This chapter describes how to set up your users if your organization distributes AWS Elemental MediaLive devices to users of AWS Elemental MediaLive, but people in your organization don't use MediaLive for encoding. Your users must have permissions to use MediaLive to transfer devices to MediaLive users.

Topics

Creating an administrator IAM user

The procedures in this section show how to create an IAM user that has full read/write administrator permissions. This administrator might be you or another person. You set up an administrator by creating a group, and then creating a user that belongs to that group:

- If your organization is new to AWS, follow both steps in this procedure: create the group, and then create the users for that group.
- If your organization is not new to AWS, then the group probably has already been created. Follow only the second step to create users for that group.

To create a full-access administrator group

1. Use your AWS account email address and password to sign in to the AWS Management Console as the AWS account root user.
2. Open the IAM console at https://console.aws.amazon.com/iam/.
3. In the navigation pane, choose **Groups**, and then choose **Create New Group**.
4. On the **Set Group Name** page, for **Group Name**, enter a name such as **`Administrators`**. Choose **Next Step**.
5. On the **Attach Policy** page, choose **Filter: Policy Type**, and then choose **Job function**.
6. In the policy list, select the check box for **AdministratorAccess**, and then choose **Next Step**.
7. On the **Review** page, review the information, and then choose **Create Group**.

Now that you have an administrator group, you are ready to create an IAM user and add the user to your group.

To add an IAM user to the full-access administrator group

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Users**, and then choose **Add user**.
3. On the **Add User** page, for **User name**, enter a name such as **`Administrator`** or **`Admin_2`** (if **`Administrator`** has already been created).
4. For **Access type**, select **AWS Management Console access**.

   For **Console password**, choose **Custom password**, and then enter a password.

If this administrator is not you, we recommend that you select **Require password reset**.

5. Choose **Next: Permissions**.
6. On the **Set permissions** page, choose **Add user to group**.
7. Select the check box for the group that you created in the preceding procedure, and then choose **Next: Review**.
8. Review the information, and then choose **Create user**. To return to the navigation pane, choose **Close**.

After you create this IAM user with administrator permissions, sign out and sign in again using the administrator credentials.

We highly recommend that from this point forward, you always sign in using the IAM administrator credentials instead of your root user credentials, unless AWS requires you to use your root user credentials to perform certain operations. For more information, see AWS Tasks That Require AWS Account Root User Credentials.

Repeat the procedure to set up more administrators (as backups), if needed. Or anyone who is now set up as a full-access administrator can set up more administrators.

Creating a non-administrator user

This section describes how to create non-administrator users and give those users limited permissions. Any person who is an administrator can perform this procedure.

You might have created some users and assigned policies to them. This procedure has a different approach—it describes how to create a group that contains a policy, then add users to that group. This approach ensures that all users are set up with the same permissions (the permissions of the group), which is particularly important if you later add more than one policy to the group.

Summary of Steps

You should create non-administrator users and give them only the permissions that they need in order to work with AWS Elemental Link devices:

- Create a custom policy that gives users the permission that they need in order to work with devices.
- Create a group for your users, and attach the policy.
- Create each user and add them to the group.

Initial setup

The first time you perform this setup, you must create a custom policy, create a group, and then create users and add them to the group.

Subsequent setup

After you have performed the initial setup, then each time you want to add a user, you only need to create the user and add them to the existing group.

To create a custom policy

- In IAM, use the IAM policy generator to create a custom policy. The policy generator lets you choose the service from a list, and then choose operations from a list. As a best practice, give the policy a name that starts with the service name, `medialive`. In the policy, include these MediaLive operations:

- `AcceptInputDeviceTransfer`
- `CancelInputDeviceTransfer`
- `ListInputDeviceTransfers`
- `RejectInputDeviceTransfer`
- `TransferInputDevice`

To create the policy, follow the prompts on the console. For detailed instructions for creating a policy, see *IAM User Guide*.

To create a group

1. In IAM, choose **Groups**, and then use the **Create New Group Wizard** to create one group that all of your users will belong to. See https://docs.aws.amazon.com/IAM/latest/UserGuide/id_groups_create.html and follow the steps for creating groups using the console.

   As a best practice, give the group a name that starts with the service name, `medialive`.
2. The **Create New Group Wizard** includes a step for attaching policies to the group as you create it. Make sure to attach the policy that you created.

To create a user

This procedure describes how to set up the user for console access, but not for AWS CLI or AWS SDK access. To set up for programmatic access, see *IAM User Guide*.

1. Make sure that you know which group that you want to add each user to. Make sure that you have already created this group (the section called “Step 4: Create the groups” (p. 41)).
2. Sign in to the AWS Management Console as an administrator, and open the IAM console at https://console.aws.amazon.com/iam/.
3. In the navigation pane, choose **Users**, and then choose **Add user**.
4. On the **Add User** page, for **User name**, enter a name for the user. Complete the following fields:
   - For **Access type**, select **AWS Management Console access**.
   - For **Console password**, select **Custom password** and enter a password.
   - For **Require password reset**, we recommend that you select the check box.
5. Choose **Next: Permissions**.
6. On the **Set permissions for user** page, choose **Add user to group**.
7. Select the check box for the appropriate group for this user, and then choose **Next: Tags**.
8. Choose **Create user**.
9. Optionally, choose **Send email** to send an email to this user. Your local email client opens with a draft email that includes the user name and sign-in URL.
10. Choose **Close** to return to the navigation pane.
11. Provide the user with their password (it is not included in the generated email). You must provide the password in a way that complies with your organization's security guidelines.

Getting started 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 *Resources: MediaLive workflow wizard* (p. 309).

  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 below.

  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 these sections:
  - The sections that are prefixed with *Setting up*.
  - 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

Getting started 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 *Setting up: IAM permissions* (p. 14), 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” (p. 64) 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” (p. 68).

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" (p. 64). 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" (p. 38).
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" (p. 64). 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" (p. 66).
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" (p. 64). Set them up to push to the two destinations that you noted in the section called "Step 3: Create an input" (p. 65). 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**.

Components of AWS Elemental MediaLive

The key building blocks of AWS Elemental MediaLive are *inputs*, *channels*, and *input security groups*. A channel in turn consists of output groups, which contain outputs, which contain video, audio, and captions "encodes."

When a channel is started (run), AWS Elemental MediaLive ingests the input. It then transcodes that video (and the related audio, captions, and metadata) and creates output assets. The information about how to transcode a given input is contained in a channel.

An input security group (p. 243) is a mechanism to prevent unauthorized third parties from pushing content into a channel that is associated with a "push" input.

Inputs

An *input* is a video asset that is to be transcoded and packaged. It may be associated with an input security group, which provides protection to the input, and with a channel, which provides details about the transcoding and packaging to perform.

AWS Elemental MediaLive supports different types of stream and file inputs (for example, RTP and HLS). The service also provides two ways to ingest the inputs, either through a push model or a pull model. For more information, see the section called “Supported input formats and protocols” (p. 529).

Channels

In MediaLive, a *channel* is attached to one or more inputs (video sources). If the channel is attached to more than one input, the inputs are processed one after the other. A channel contains the details that instruct MediaLive how to transcode (decode and encode) and package the inputs into specific outputs. The key components of a channel are an encode, an output, and an output group.

Encodes

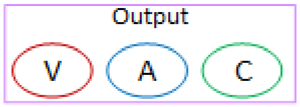
An *encode* is the smallest component on the output side of a channel. Each encode contains the instructions for one video asset, one audio asset, or one captions asset that will be created by the transcoding process. Different encodes have different characteristics. For example, one video encode produced from the input might be high resolution while another is low resolution. Or one audio encode might use the AAC audio codec while another uses the Dolby Digital audio codec.

A channel can contain multiple video, audio, and captions encodes.

In the following illustration, the red circle represents a video output, the blue circle represents an audio output, and the green circle represents a captions output.

Outputs

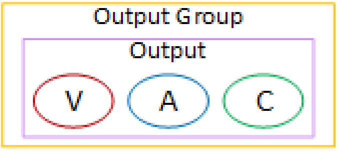
An *output* contains the encodes that belong together. For example, one output will contain the combination of video, audio, and captions encodes that make sense for one purpose, while another output will contain a different combination.



The output holds packaging instructions that apply to all the encodes in that output. For example, the packaging instructions for a UDP output are different from those for an Archive output. The encodes inside the outputs might be the same or different. But the packaging instructions are different.

Output groups

An *output group* contains related outputs. An output group might contain only one output or it might contain several outputs. The output group holds details about the destination for all the outputs in that group.



Input security groups

An *input security group* is a group that you create and associate with specific input types, to prevent unauthorized third parties from pushing content into a channel. For more information, see *Resources: MediaLive input security groups* (p. 243).

How components are associated

The association between inputs and a channel is defined in the channel. In other words, to associate a channel with one or more inputs, you set up the channel to point to those inputs.

After you create this association, you can do the following:

- View the channel details to identify the associated input.
- View the input details to identify the associated channel.

The association between an input and an input security group is defined in the input. In other words, to associate an input with an input security group (or with several groups), you set up the input to point to a specific input security group (or groups).

After you create this association, you can do the following:

- View the channel details to identify the associated input.
- View the input details to identify the associated channel.

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* (p. 121) 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” (p. 73).
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” (p. 75).
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” (p. 76).
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” (p. 78).

   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” (p. 83).
6. Arrange for setup with the upstream system so that it can connect to MediaLive. Also, create the MediaLive input so that MediaLive can connect to the upstream system. See the section called “Step 6: Coordinate with upstream system and create inputs” (p. 87).
7. Coordinate with the downstream system or systems to provide a destination for the output groups that MediaLive will produce. See the section called “Step 7: Coordinate with downstream systems” (p. 111).

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* (p. 4)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 (p. 78). 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” (p. 503).
   - 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 containers and downstream systems” (p. 535).
4. If your preferred downstream system is another AWS media service, read this for information about choosing the service (p. 74).
5. If your downstream system supports Microsoft Smooth Streaming, see the section called “Options for Microsoft Smooth” (p. 75) 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 (p. 131).

   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 (p. 74).

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.

There are differences in the setup that you must follow for each type of output group:

- Read the information about channel class (p. 445) and decide whether you will create a standard channel or a single-pipeline channel. If you decide to create a single-pipeline channel, then you should create an HLS 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.

- 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” (p. 387).

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 (p. 78). Decision making in a workflow starts with the downstream system, then works back to the upstream system.

Result of this procedure

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.

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 for outputs” (p. 537). 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 for outputs” (p. 537). 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 (p. 82).
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” (p. 483).

   .

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” (p. 445).

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” (p. 351).

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 the section called "How MediaLive Works" (p. 1).

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" (p. 78)
   - the section called "Assess video content" (p. 80)
   - the section called "Assess audio content" (p. 81)
   - the section called "Assess captions" (p. 82)
2. Make a note of the following three characteristics of the source stream. You will need this information when you set up the channel (p. 147):

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

At the end of this step, you will know the following:

- You will be confident that MediaLive can ingest the content.
- You will know what type of MediaLive input you will create to ingest the source content.
- You will have the information that you need to extract the video, audio, and captions from the source (from the MediaLive input).
- You will have the basic information for setting up the channel to transcode the content.

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 (p. 76), 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.

Information to obtain	Verify the following
	• 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. Make sure that this format is listed in the table in the section called “Supported input formats and protocols” (p. 529). If it is not 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 inputs” (p. 532).
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” (p. 79). 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 “Supported codecs for inputs” (p. 533). 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 to handle the anticipated maximum bitrate of the source content. If you are setting up standard channels (to implement pipeline redundancy (p. 76)), make sure that the bandwidth is double the anticipated 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 characteristics 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 "Supported codecs for inputs" (p. 533).
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 (p. 81).
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 (p. 73) for your workflow.

1. Go to the section called “Reference: supported captions” (p. 514) and find the section for the output group. For example, find the section called “HLS or MediaPackage output” (p. 522). 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 (p. 75). 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.

Result of this step

As a result of this assessment, you should have a list of the video, audio, and captions assets that the content provider supports.

Example

Information	Format	Characteristics
Source formats and packaging	RTP	with FEC
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

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

Information	Format	Characteristics	Identifiers
	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

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 (p. 82) 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 (p. 82) 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 Working with MediaLive devices*Resources: MediaLive devices* (p. 215).
- 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 “Timecode configuration” (p. 481).
- 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” (p. 453).

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: Coordinate with upstream system and create inputs

After you have identified the upstream system and content source for the channel, 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 is different for each combination of upstream system (format and delivery protocol) and input type. You identified the upstream system and input type when you assessed the upstream system (p. 78).

Result of this step

At the end of this step, you will have completed all the steps to prepare the upstream system to deliver content to MediaLive.

Topics

- Setting up an RTMP push source (p. 99)
- Setting up an RTMP VPC source (p. 102)
- Setting up an RTP push source (p. 104)
- Setting up an RTP VPC source (p. 107)

Setting up a CDI source

This section describes how to set up content on an upstream system that is using AWS Cloud Digital Interface (AWS CDI) to deliver uncompressed content over 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 AWS Elemental MediaLive.

With a CDI source, the upstream system *pushes* the content to MediaLive.

This diagram illustrates the final result of this setup.

To perform this setup, you must work with an Amazon VPC user, and with an operator at the upstream system.

Topics

- Request setup on the VPC (p. 88)
- Create a CDI input (p. 89)
- Ensure correct setup on the upstream system (p. 89)
- Result of this procedure (p. 90)

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. You need two subnets because a CDI input is always a standard-class input (p. 386), even if your channel is a single-pipeline channel. For information about input classes, see the section called “Channel classes and input classes” (p. 386).

     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`

Create a CDI input

You must create a CDI input to connect the upstream system to MediaLive. Use the information about the VPC that you obtained from the Amazon VPC user:

- The ID of the VPC
- The IDs of the two subnets

- The IDs of the security groups for the subnet or subnets

For details on creating a CDI input, see the section called “CDI input” (p. 221).

Ensure correct setup on the upstream system

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 (p. 76), 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.
    - 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 (p. 90). These URLs each have a private IP address from the subnet range, and they specify port 5000. For example:

      ```
      10.99.39.23:5000
      ```

      ```
      192.0.2.54:5000
      ```

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 (p. 90).
    - 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:

      ```
      10.99.39.23:5000
      ```

      ```
      192.0.2.54:5000
      ```

To set up for a single-pipeline channel

If the planned channel is a single-pipeline channel (p. 76), you can request that the operator at the upstream system provides only one source, even though the CDI input is a standard-class input. The content from the upstream system will flow as follows:

- 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 boxes in the diagram after this procedure (p. 90). The URL has a private IP address from the subnet range, and it specifies port 5000.

   ```
   10.99.39.23:5000
   ```
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 (p. 90).
   - Push to the correct URL on MediaLive. For example, they must push to:

   ```
   10.99.39.23:5000
   ```

Result of this procedure

As a result of this setup, a CDI 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 both endpoints (if you are setting up a standard channel) or to one endpoint (if you are setting up 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 CDI 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 AWS Elemental Link source

This section describes how to set up content that is being delivered from an AWS Elemental Link device, and how to create an Elemental Link input that connects the device to MediaLive.

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

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" (p. 386).
- 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.

Create an Elemental Link input

You must create an Elemental Link input to connect the AWS Elemental Link device to MediaLive. To create an Elemental Link input, see the section called "Elemental Link input" (p. 224).

Result of this procedure

As a result of this setup, an Elemental Link input exists that identifies the AWS Elemental Link device or devices 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 source

This section describes how to set up the source content on the upstream system, and how to create an HLS input that connects the upstream system to MediaLive.

With an HLS input, MediaLive connects to the upstream system when the channel starts and *pulls* the sources. The upstream system is typically a server that acts as a location for the source content. This server is not the camera or encoder at the start of the workflow.

This diagram illustrates the final result of this setup.

To perform this setup, you must work with an operator at the upstream system.

Topics

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" (p. 386).

  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 (p. 78). 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 (p. 79) 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.

You need this information to create the HLS input.

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`

Create an HLS input

You must create an input to connect MediaLive to the content source on the upstream server. To create an HLS input, see the section called "HLS input" (p. 225).

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 earlier in this section (p. 92), and that you configured into the HLS input. They correspond to the URLs shown in the diagram after this procedure (p. 93).

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.

At runtime of the channel, MediaLive will connect to the two URLs (for a standard channel) or the single URL (for a single-pipeline channel), and pull the HLS manifests into MediaLive.

Setting up for a MediaConnect source

This section describes how to create flows in AWS Elemental MediaConnect to deliver source content from the upstream system to AWS Elemental MediaLive, and how to create a MediaConnect input that connects the upstream system to MediaLive.

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.)

This diagram illustrates the final result of this setup.

To perform this setup, you must work with an AWS Elemental MediaConnect user.

Topics

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 MediaConnect input (p. 226).)
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 MediaConnect input (p. 226).)
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.

Create a MediaConnect input

After MediaConnect is set up, you must create an input in order to connect AWS Elemental MediaLive to the MediaConnect flows. A MediaLive user performs this step.

For instructions on creating a MediaConnect input, see the section called "MediaConnect input" (p. 226).

Result of this procedure

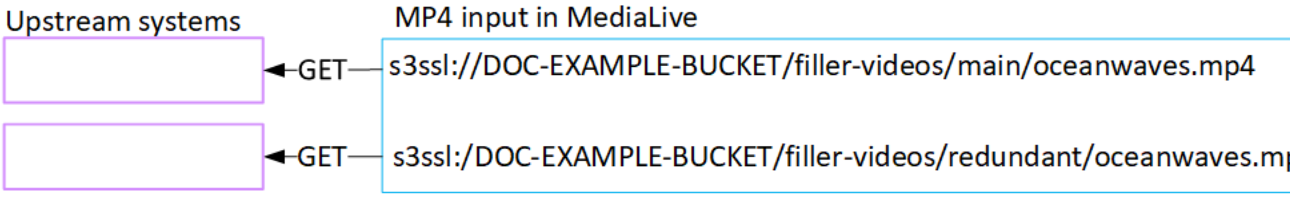
As a result of this setup, one or two MediaConnect flows exist. Each flow has a source that the upstream system is pushing to. Each flow also has one output for the use of MediaLive. A MediaLive input exists that specifies the ARNs for those outputs.

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 source

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. With an MP4 input, MediaLive connects to the upstream system when the channel starts and *pulls* the sources.

This diagram illustrates the final result of this setup.


Upstream systems
MP4 input in MediaLive
GET
s3ssl://DOC-EXAMPLE-BUCKET/filler-videos/main/oceanwaves.mp4
GET
s3ssl:/DOC-EXAMPLE-BUCKET/filler-videos/redundant/oceanwaves.mp

To perform this setup, you must work with an operator at the upstream system.

Topics

- Obtain information (p. 96)
- Create an MP4 input (p. 97)
- Ensure correct setup on the MP4 upstream system (p. 97)
- Result of this procedure (p. 98)

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" (p. 386).

  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.

You need this information to create the MP4 input.

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.

Create an MP4 input

You must create an input to connect AWS Elemental MediaLive to the content source on the upstream server. To create an MP4 input, see the section called “MP4 input” (p. 229).

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:

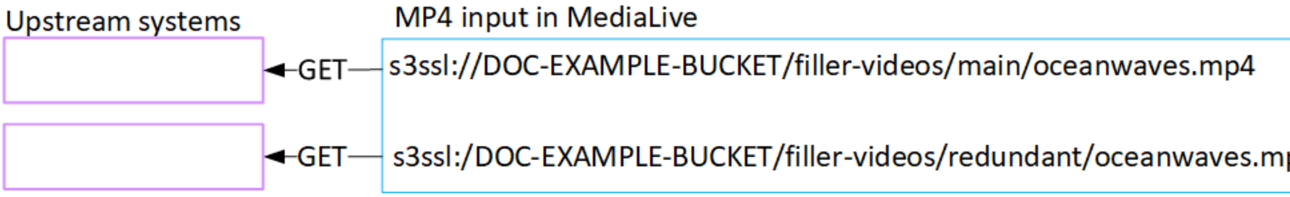
- They set up to deliver the correct number of sources:
  - If the MediaLive channel is a standard channel, the operator must set up two file sources. They must make sure that the two files are identical in terms of video resolution and bitrate.

- 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 (p. 96), and that you configured into the MP4 input. They correspond to the URLs shown in the diagram after this procedure (p. 98).

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.


Upstream systems
MP4 input in MediaLive
GET
s3ssl://DOC-EXAMPLE-BUCKET/filler-videos/main/oceanwaves.mp4
GET
s3ssl:/DOC-EXAMPLE-BUCKET/filler-videos/redundant/oceanwaves.mp

Setting up an RTMP pull source

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. With an RTMP Pull input, MediaLive connects to the upstream system when the channel starts and *pulls* the sources.

With an RTMP pull input, the upstream system is typically an upstream server that serves as a location for the source content. The server is not the camera or encoder at the start of the workflow.

This diagram illustrates the final result of this setup.

To perform this setup, you must work with an operator at the upstream system.

Topics

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” (p. 386). For information about input classes and their uses, see the section called “Channel classes and input classes” (p. 386).

  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.

You need this information to create the RTMP input.

Create an RTMP pull input

You must create an input to connect AWS Elemental MediaLive to the content source on the upstream server. To create an RTMP Pull input, see the section called “RTMP pull input” (p. 230).

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 (p. 96), and that you configured into the RTMP input. They correspond to the URLs shown in the diagram after this procedure (p. 102).

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 source

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.

With an RTMP Push input, the upstream system *pushes* the content to MediaLive.

This diagram illustrates the final result of this setup.

To perform this setup, you must work with an operator at the upstream system.

Topics

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" (p. 386). For information about input classes and their uses, see the section called "Channel classes and input classes" (p. 386).

  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 (p. 102).

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" (p. 243).

Create an RTMP push input

After you create the input security group, you must create an input to connect AWS Elemental MediaLive to the content source on the upstream system.

For details on creating an RTMP push input, see the section called “RTMP push input” (p. 231).

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 (p. 76).

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 (p. 102). 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 source

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.

With an RTMP Push input, the upstream system *pushes* the content to MediaLive.

This diagram illustrates the final result of this setup.

To perform this setup, you must work with an Amazon VPC user, and with an operator at the upstream system.

Topics

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`

Create an RTMP input

You must create an input to connect AWS Elemental MediaLive to the content source on the upstream system. Use the information that you obtained from the Amazon VPC user:

- The ID of the VPC
- The IDs of the two subnets
- The IDs of the security groups for the subnet or subnets

For details on creating an RTMP VPC input, see the section called “RTMP VPC input” (p. 233).

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 (p. 76).

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 (p. 104). 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 (p. 104).
   - 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 (p. 76).

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 (p. 104). 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 (p. 104).
   - 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.

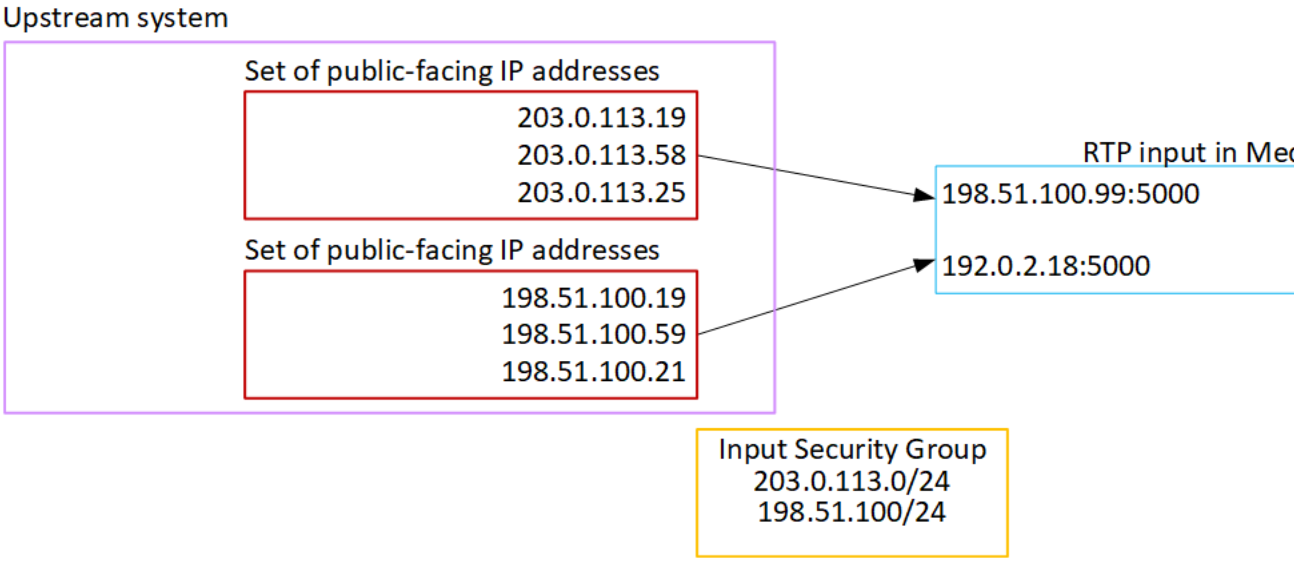
At runtime of the channel, MediaLive reacts to the content that is being pushed and ingests it.

Setting up an RTP push source

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.

This diagram illustrates the final result of this setup.



To perform this setup, you must work with an operator at the upstream system.

Topics

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 (p. 386), even if your channel is a single-pipeline channel. For information about input classes, see the section called “Channel classes and input classes” (p. 386).

  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`

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” (p. 243).

Create an RTP Input

After you create the input security group, you must create an input to connect AWS Elemental MediaLive to the content source on the upstream system.

For details on creating an RTP push input, see the section called “RTP input” (p. 236).

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 (p. 76).

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 (p. 107). 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 (p. 76).

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 (p. 107). 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:

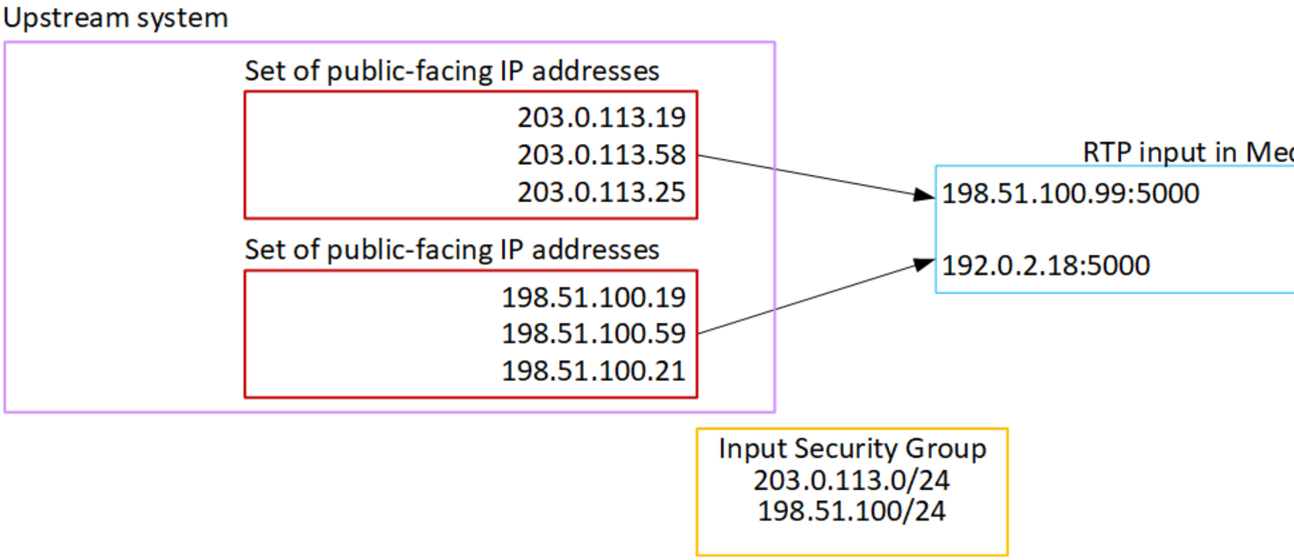
  ```
  198.51.100.99:5000
  ```
- Send over RTP, not UDP. The UDP protocol is not supported for an input into MediaLive.

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.

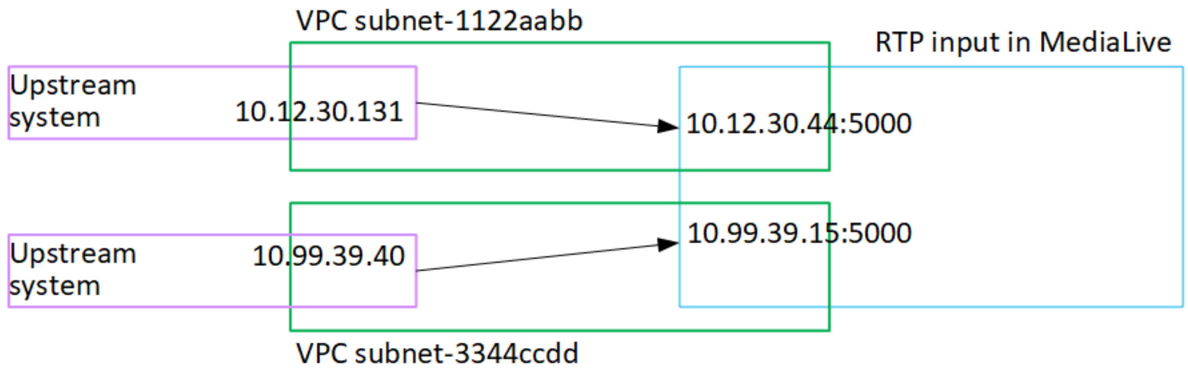


Setting up an RTP VPC source

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.

This diagram illustrates the final result of this setup.



To perform this setup, you must work with an Amazon VPC user, and with an operator at the upstream system.

Topics

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 (p. 386), even if your channel is a single-pipeline channel. For information about input classes, see the section called “Channel classes and input classes” (p. 386).
     - 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`

Create an input for RTP push from Amazon VPC

You must create an input to connect AWS Elemental MediaLive to the content source on the upstream system. Use the information that you obtained from the Amazon VPC user:

- The ID of the VPC
- The IDs of the two subnets
- The IDs of the security groups for the subnet or subnets

For details on creating an RTP VPC input, see the section called "RTP VPC input" (p. 238).

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 (p. 76).

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 (p. 110). 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 (p. 110).
   - 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 (p. 76).

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 (p. 110). 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 (p. 110).
   - 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.

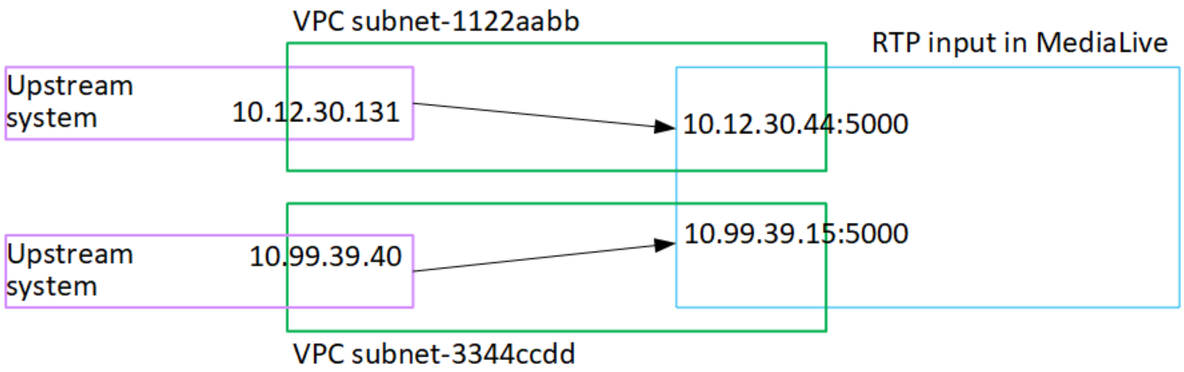
Result of this procedure

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.



Step 7: 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" (p. 112)
Frame capture	Amazon S3	the section called "Archive or frame capture" (p. 112)
HLS	Amazon S3	the section called "HLS to Amazon S3" (p. 113)
HLS	AWS Elemental MediaStore	the section called "HLS to MediaStore" (p. 114)
HLS	AWS Elemental MediaPackage	the section called "HLS to MediaPackage" (p. 115)
HLS	HTTP server or Akamai CDN	the section called "HLS to HTTP" (p. 116)
MediaPackage	AWS Elemental MediaPackage	the section called "MediaPackage " (p. 117)
Microsoft Smooth	HTTP server	the section called "Microsoft Smooth" (p. 118)
RTMP	RTMP server	the section called "RTMP" (p. 119)
UDP	UDP address	the section called "UDP" (p. 119)

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

Follow this procedure if you determined (p. 73) 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 (p. 158).

You must follow this procedure for each output group.

To arrange setup of the destination

1. Decide if you need two destinations for the output:
   - You need two destinations for output from a standard channel (p. 76).
   - You need one destination for output from 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” (p. 113), 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” (p. 159) or the section called “Destination fields” (p. 165).

   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” (p. 56).

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 (p. 159).

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.

HLS output group to Amazon S3

Follow this procedure if you determined (p. 73) 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 (p. 168).

You must follow the procedure for each HLS output group.

To arrange setup of the destination

1. Decide if you need two destinations for the output:
   - You need two destinations for output from a standard channel (p. 76).
   - You need one destination for output from 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” (p. 114), 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” (p. 182).

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” (p. 56).

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 (p. 172).

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 (p. 73) 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 (p. 168).

You must follow the procedure for each HLS output group.

To arrange setup of the destination

1. Decide if you need two destinations for the output:

   - You need two destinations for output from a standard channel (p. 76).

- You need one destination for output from 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" (p. 182). 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" (p. 56).

HLS output group to MediaPackage

Follow this procedure if you determined (p. 73) that you will create an HLS output group, and will send to AWS Elemental MediaPackage over HTTPS.

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 (p. 168).

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" (p. 74) for a description of the differences. This section describes the second option.

Follow this guidance for each HLS output group.

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 (p. 76) (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, not the channel name.

- 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 HTTP

Follow this procedure if you determined (p. 73) 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 (p. 168).

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).

You must follow the procedure for each HLS output group.

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 for output from a standard channel (p. 76).
   - You need one destination for output from 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” (p. 422).
6. If you are setting up a standard channel (p. 76), 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” (p. 426), and specifically the section called “Rules for most systems” (p. 429) and the section called “Rules for Akamai” (p. 430) 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.

      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 (p. 73) 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 (p. 192).

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" (p. 74) for a description of the differences. This section describes the first option.

Follow this guidance for each MediaPackage output group.

To arrange setup of the destination

1. Ask the MediaPackage user to create one channel. Even if the MediaLive channel is a standard channel (p. 76) (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" (p. 56).

Microsoft Smooth output group

Follow this procedure if you have determined (p. 73) 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 (p. 195).

You must follow the procedure for each Microsoft Smooth output group.

To arrange setup of the destination

1. Decide if you need two destinations for the output:
   - You need two destinations for output from a standard channel (p. 76).
   - You need one destination for output from 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 (p. 73) 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 (p. 199).

You must follow the procedure for each RTMP output, if the RTMP output group has more than one output. Each output within the group has its own destination.

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 (p. 76), 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 (p. 73) 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 (p. 202).

You must follow the procedure for each UDP output group.

To arrange setup of the destination

1. Decide if you need two destinations for the output:

   - If the MediaLive channel is a standard channel (p. 76), 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, starting from the outputs your required. You then worked back to the source—assessing the upstream system, obtaining information about the sources, and setting up the source inputs. You then returned to the output side, coordinating with the downstream system for delivery of the outputs that MediaLive will produce.

As part of this planning, you have already created some of the resources that you need for the workflow:

- Inputs for the sources
- Input security groups, for some inputs.

You are now ready to design the channel that is the key resource in your workflow. See *Setup: Planning the channel* (p. 121).

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 (p. 72).

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 (p. 73) 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 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.
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 (p. 121).
   - 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 (p. 121).

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.
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.
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 (p. 122).

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.
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 each caption format belongs to. See the list in the section called “Captions categories” (p. 517). 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.
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, 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 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” (p. 340).

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” (p. 484).

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
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 (p. 121). 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
	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

Output group	Type of encode	Encode nickname	Characteristics of the encode
		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 “Result of this step” (p. 83).
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
		CaptionsB	WebVTT (object-style) converted from embedded, in French	Embedded	Channel 2

Output group	Type of encode	Encode nickname	Characteristics of the encode	Source	Identifier in source
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
		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 (p. 153).
- You will use the characteristics information when you create the encodes (p. 204) 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

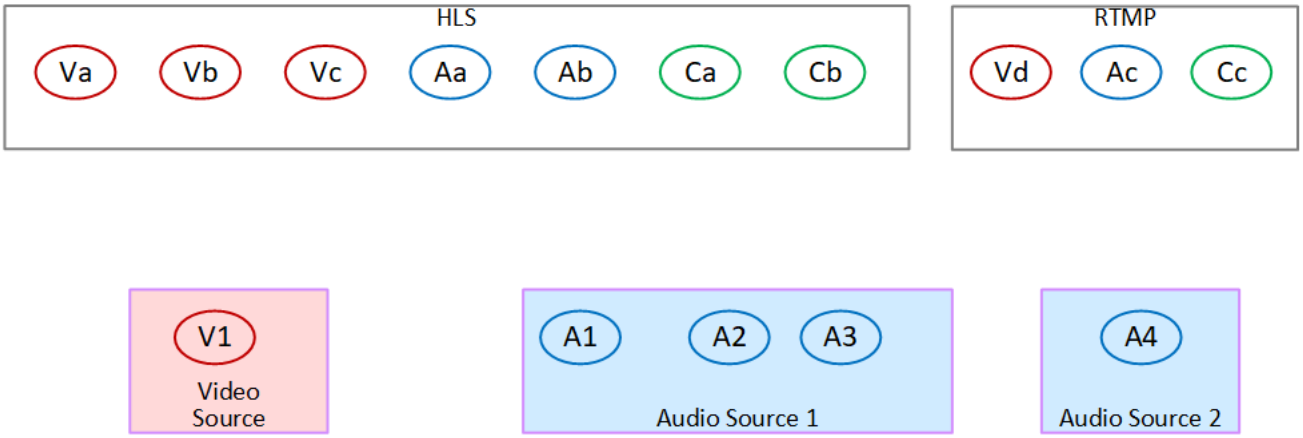
Input asset	Asset nickname	Source	Characteristics	Identifier in Source
video 1	Video1	Video	HEVC	PID 600
audio 1	Audio1	Audio	AAC 2.0 Spanish	PID 746
audio 2	Audio2		AAC 2.0 French	PID 747

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

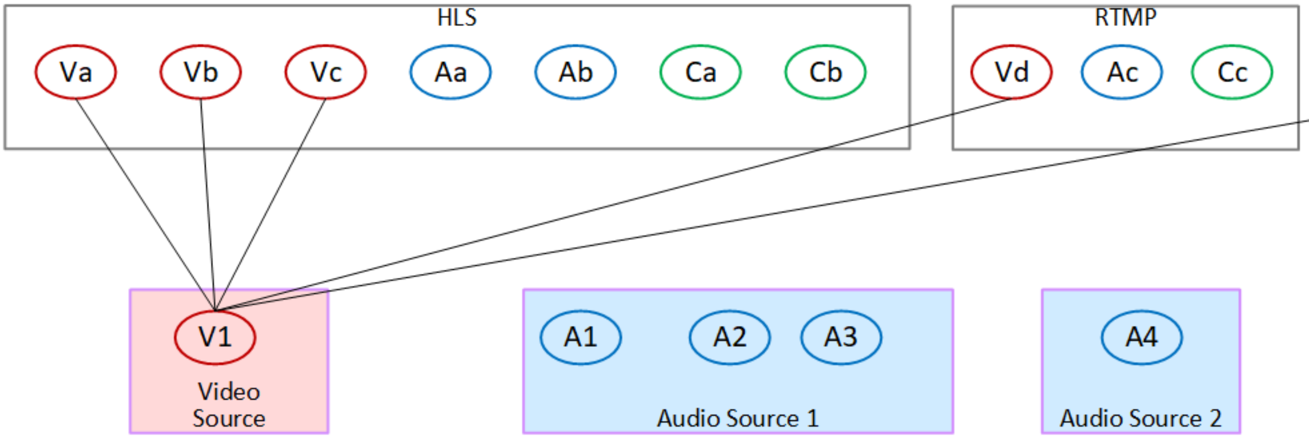
Example of mapping

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.

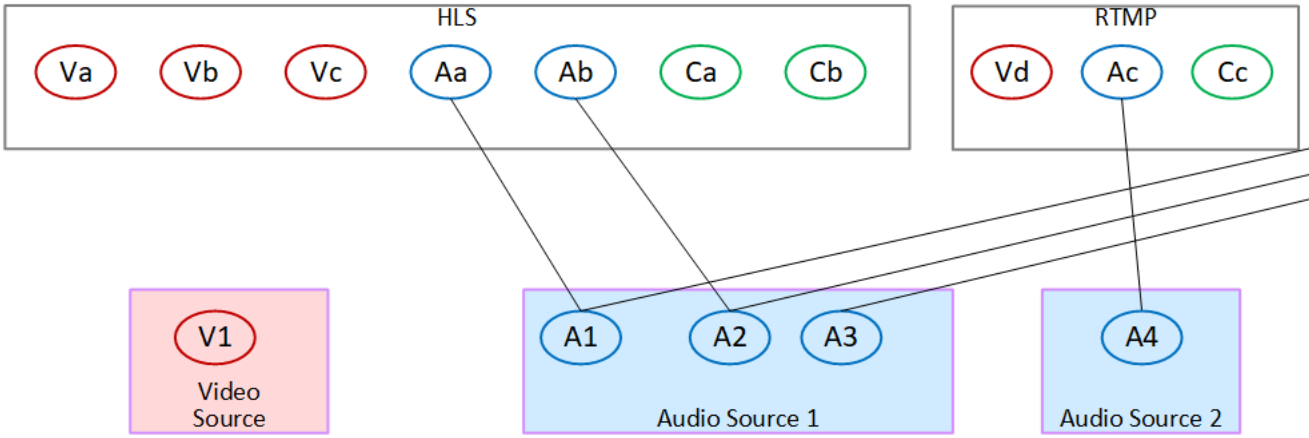
Encodes and assets



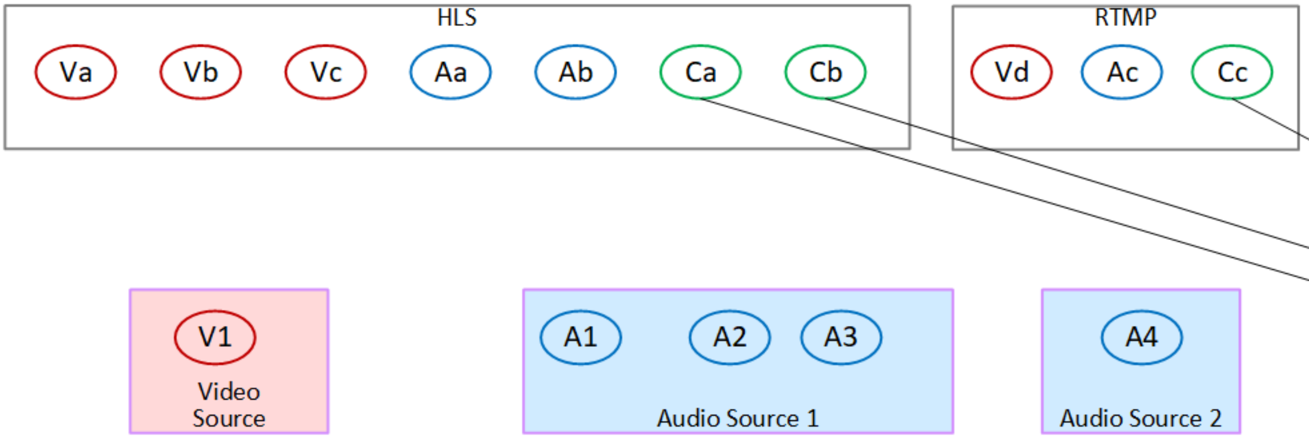
Mapping video encodes to assets



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 (p. 121) 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.

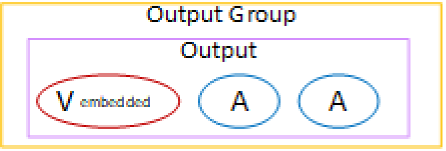
Topics

Organize encodes in an Archive output group

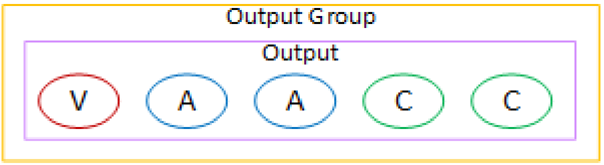
An Archive output group can contain one video encode, one or more audio encodes, and one or more captions encodes (either embedded or object-style) (p. 517).

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 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 (p. 75) from your downstream system.

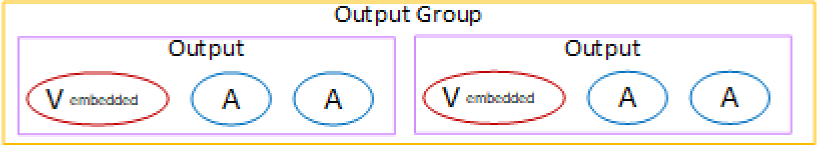
Downstream players that require bundled audio

Plan for the output group to contain the following:

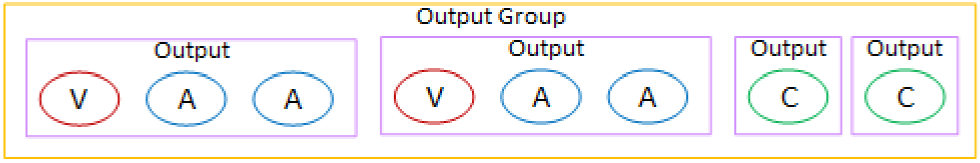
- One output for each video encode. This output holds one video encode, all the audio encodes, and all the captions encodes (if the captions are embedded).

  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

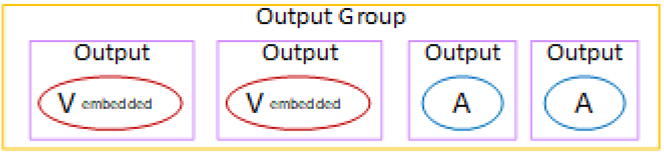
Plan for the output group to contain the following:

- 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.

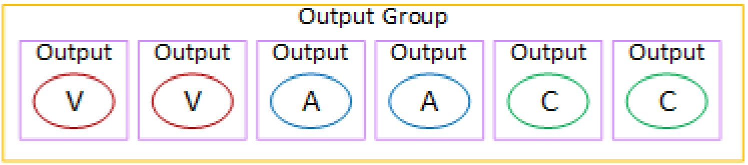
  The audio encodes might be for different languages, or they might be for different bitrates, or they might be for different languages and bitrates.
- One output for each captions encode, if the captions are sidecars.

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 (p. 75) from your downstream system.

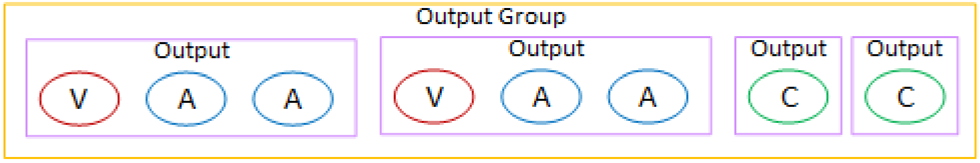
Downstream players that require bundled audio

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

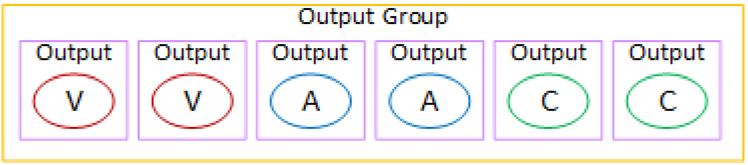
Plan for the output group to contain the following:

- 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.

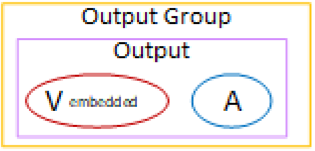


Organize encodes in an RTMP output group

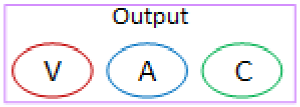
An RTMP output group contains one video encode, one audio encode, and one captions encode.

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.

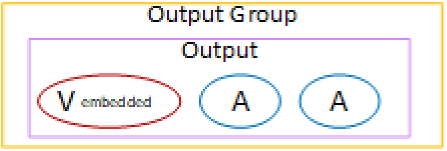


Organize encodes in a UDP output group

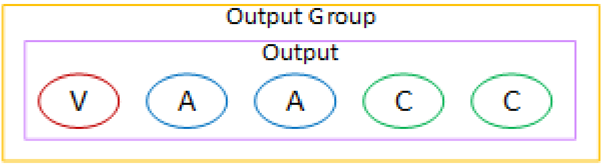
A UDP output group can contain one video encode, one or more audio encodes, and one or more captions encodes (either embedded or object-style).

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.



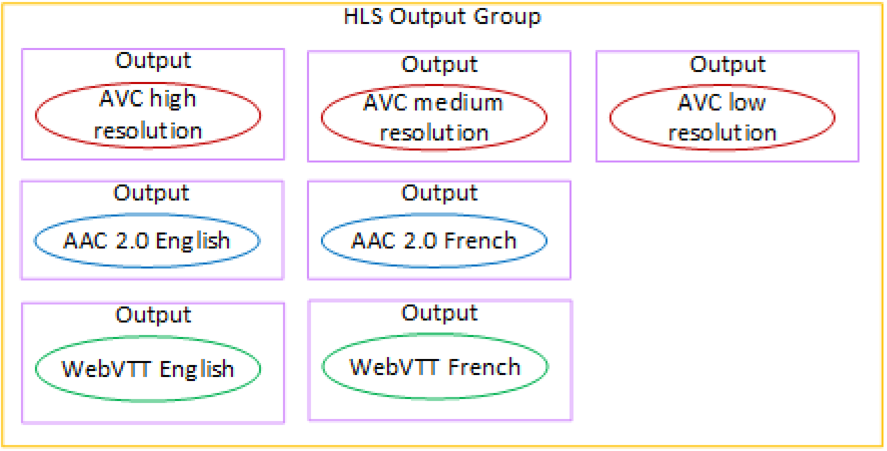
Examples

the section called “Example of a plan” (p. 124) 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.

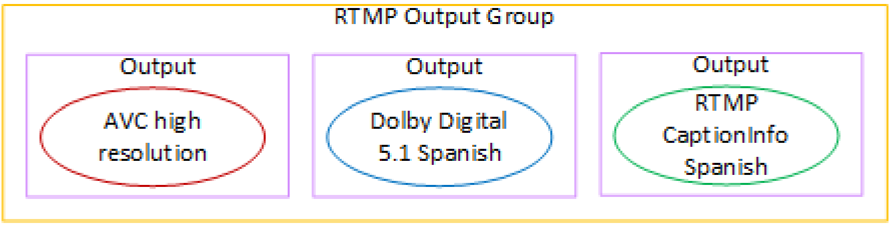
HLS output group

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.



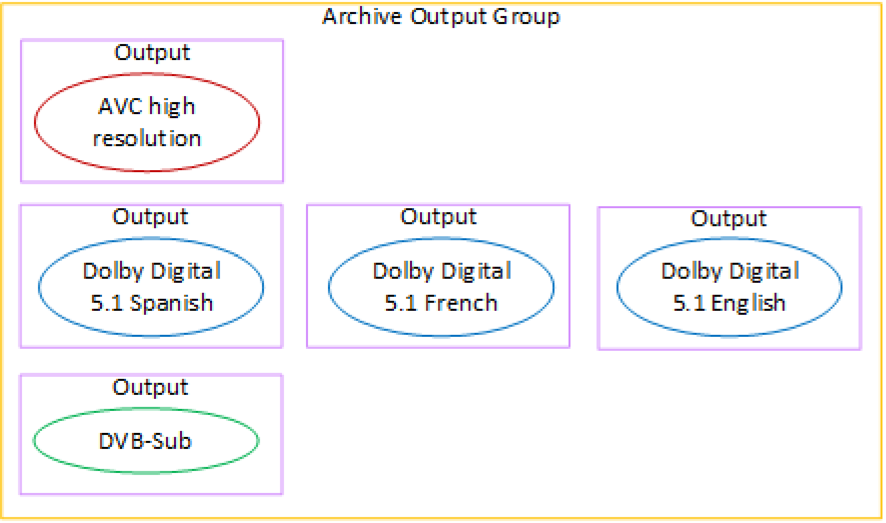
RTMP output group

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 (p. 121) 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

In the section called “Step 2: Map outputs to sources” (p. 125), 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 from embedded, in English	Embedded	Channel 4
		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 – color space" (p. 486).
- For information about the Additional encoding settings fields, see the section called "Video – enhanced VQ" (p. 499)
- For information about the **Rate control** fields, see the section called "Video – rate control mode" (p. 500). 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 "Timecode configuration" (p. 481).

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" (p. 476).

You will use encode sharing and encode cloning when you create the encodes in the channel, starting with the section called "Step 6: Set up video" (p. 204).

- 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 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.

Encode nickname	Characteristics of the encode	Source	Opportunity	Action
CaptionsA	WebVTT (object-style) converted from embedded, in English	Embedded		Create this encode from scratch.
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
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 configuration 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.
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 (p. 73).

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 channels. See *Resources: MediaLive channels* (p. 144).

Working with AWS Elemental MediaLive channels

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. 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.

Before you start to create a channel, you should plan your channel (p. 121) to identify the following elements:

- Inputs that the channel will use
- Output groups for the channel
- Outputs (within each output group)
- Video, audio, and captions encodes (in each output) that the channel will produce

There are three ways to create a channel:

- Create from scratch.
- Use a built-in or custom template.
- Clone an existing channel.

Once 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

A channel contains the details that instruct AWS Elemental MediaLive about how to transcode (decode and encode) and package your input into specific outputs.

Before you start the process of creating a channel, you should plan your upstream and downstream systems (p. 72) and plan your channel (p. 121) to identify the following elements:

- Inputs that the channel will use
- Output groups for the channel
- Outputs (within each output group)
- Video, audio, and captions encodes (in each output) that the channel will produce

There are three ways to create a channel:

- **Create from scratch.** See the topics (steps 1-9) in this chapter.
- **Use a built-in or custom template.** See the section called "Creating a channel from a template or by cloning" (p. 210).
- **Clone an existing channel.** See the section called "Creating a channel from a template or by cloning" (p. 210).

> **Note**
> For information about additional steps for setting up a channel for use in a multiplex program, see the section called "Step 5: Create the channels" (p. 436).

Topics

Getting ready

We recommend that before you start creating the channel, you plan the workflow (p. 72) and plan the channel (p. 121). In both these planning procedures, you obtain information that you need to create the channel.

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 "Step 1: Complete channel details" (p. 146):

  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* (p. 72).
- You need the following information in the section called "Step 2: Attach inputs" (p. 148):

  The names of the input or inputs to use in this channel. You created the input or inputs as part of step 6 of *Setup: Preparing upstream and downstream* (p. 72).
- You need the following information to create the input selectors, as part of the procedure in the section called "Step 3: Complete input settings" (p. 152):
  - The assets to extract from each input. You identified these assets in the section called "Step 2: Map outputs to sources" (p. 125), as part of planning the channel.
- You need the following information in the section called "Step 5: Create output groups" (p. 158):
  - The output groups to create. You should have identified these output groups in step 1 of *Setup: Preparing upstream and downstream* (p. 72).
  - The outputs to create. You should have designed the outputs and encodes (video, audio,and captions) when you planned the channel (p. 121).
  - Information about the destinations for the outputs of each output group. You obtained this information in step 7 of *Setup: Preparing upstream and downstream* (p. 72).

- You need the following information in the three steps that start with the section called “Step 6: Set up video” (p. 204):
  - Details about the output encodes (video, audio, and captions) to create in each output group. You made these decisions in *Setup: Planning the channel* (p. 121).

Step 1: Complete the channel details

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

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 input (p. 219) 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” (p. 146).
   - For information about the **Channel template** section, see the section called “Creating a channel from a template or by cloning” (p. 210).
   - In **Channel class**, choose the class. See the section called “Channel class” (p. 147).
   - In **Input specifications**, complete the fields to match your input. See the section called “Input specifications settings” (p. 147).
   - In the **Tags** section, create tags if you want to associate tags with this channel. For more information, see the section called “Tagging resources” (p. 479).
6. When ready, go to the next step (p. 148).

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 AWS IAM administrator about your 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 (p. 445), 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 (p. 78) 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.
   - 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 for determining 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.

Step 2: Attach inputs to the channel

When you planned your workflow, you should have coordinated with the upstream system for delivery of the source content. As part of that procedure (p. 87), you created an input for each content source.

You must now attach the input 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” (p. 402). 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** page, 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 ” (p. 149)

     the section called “Channel input—Elemental Link push input” (p. 149)
   - the section called “Channel input—HLS pull input” (p. 149)
   - the section called “Channel input—MediaConnect push input” (p. 150)
   - the section called “Channel input—MP4 pull input” (p. 150)
   - the section called “Channel input—RTMP push input” (p. 151)
   - the section called “Channel input—RTMP pull input” (p. 151)
   - the section called “Channel input—RTP push input” (p. 152)
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 (p. 152).

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.

Step 3: Complete the settings for each input

As soon as you attach the input on the **Attach** input pages, the **Input attachment** section closes and the **General input settings** section appears. You must complete these fields to configure 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. 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” (p. 402).
3. When ready, go to the next step (p. 157).

Topics

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 – color space" (p. 486).

Determining whether you need to create a video selector

When you planned the channel, you should have identified the video (p. 125) 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

Input type	Method of extracting video
	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.**
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 (p. 125) 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.
Elemental Link	Yes, if the input contains two mono channels. For example,	One

Input type	Need to create selector?	Number of selectors to create
	English and French. On the output side of the channel, you will need to remix the channels, in order to separate the channels. To do that, you will need a selector name. But you won't need to complete the selector settings.	
	No, if the input contains two stereo channels. For example, English stereo.	None
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
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	Leave this field blank.
HLS	Select in one of these ways:

Input type	How to complete Selector settings
	• 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.
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.
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" (p. 361), specifically the section called "Step 1: Create captions selectors in the input" (p. 367).

Step 4: Complete the general settings

AWS Elemental MediaLive has several *settings* that apply globally to all outputs. MediaLive also has *features* that are optional but that apply globally to all outputs if they are enabled.

These settings and features apply to all outputs. Therefore, they appear on the **General settings** page, rather than in individual output groups and outputs.

To complete the general settings

1. On the **Create channel** page, in the **Channel** section, 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 (p. 158).

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" (p. 453).

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" (p. 453).

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" (p. 453).

Feature activations

Optional features. You can enable the input prepare feature for input switching. For more information, see the section called "Input prepare" (p. 395).

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 change how MediaLive handles input loss.

The behavior that you configure here applies to all the inputs attached to the channel.

When MediaLive detects that the input has not arrived within the expected time, it repeats the previous frame for a configurable number of milliseconds (from zero to forever). When that time expires, it displays a black frame for a configurable number of milliseconds (from zero to forever). When that time expires, it switches to a specified slate or to a specified color. When input resumes, the normal ingest continues.

You can change this behavior: for **Input loss behavior**, choose **Input Loss Behavior**. The default values are shown in the fields that appear. Change the fields as needed.

Motion graphics configuration

Optional feature. You can enable the motion graphics overlay feature. For more information, see the section called “Motion graphics overlay” (p. 432).

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” (p. 441).

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 “Timecode configuration” (p. 481).

Logging

Optional feature. You can enable logging of activity on this individual channel. For detailed information about this feature, see the section called “Monitoring using CloudWatch Logs” (p. 330).

To enable logging, choose a log level other than **DISABLED**. The levels are listed from least to most verbose.

To disable logging, choose **DISABLED**.

Step 5: Create output groups and outputs

In this step, you create the output groups and outputs that you identified when you planned the channel (p. 121). AWS Elemental MediaLive supports different output types. For more information, see the section called “Supported codecs for outputs” (p. 537).

Topics

Creating an archive output group

When you planned the workflow for your channel (p. 73), you might have determined that you want to include an Archive output group. An Archive group always sends output to an S3 bucket.

Topics

- Fields for the output container (p. 164)
- Fields for the video, audio, and captions streams (encodes) (p. 164)

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 (p. 159).
   - **Archive settings** – This section contains fields for the output destination (p. 159).
   - **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 (p. 159) and the output container (p. 164).
   - **Stream settings** – This section contains fields for the output streams (p. 164) (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 “Step 9: Save channel” (p. 210).

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 (p. 112) 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" (p. 161).

- 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 (p. 112), 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 number starting

Portion of the URL	Example	Comment
		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 (p. 147), or one destination when it is set up as a single-pipeline channel.	`s3ssl://DOC-EXAMPLE-BUCKET/channel59/delivery/curling`
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 "Reference: identifiers for variable data" (p. 538).	`$dft$`

Portion of the destination URL	Field	Example
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" (p. 56).
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 (p. 113).
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 (p. 162).

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 "Reference: identifiers for variable data" (p. 538).
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`** 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 “Step 6: Set up video” (p. 204)
- the section called “Step 7: Set up audio” (p. 206)
- the section called “Step 8: Set up captions” (p. 208)

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 (p. 73), 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

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 (p. 165).
   - **Frame capture settings** – This section contains a field for the output group name and for the output destination (p. 165).
   - **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 (p. 165).
   - **Stream settings** – This section contains fields for the output streams (p. 168) (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 “Step 9: Save channel” (p. 210).

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 (p. 112) 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” (p. 167).

- 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 (p. 112), 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	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.

Portion of the URL	Example	Comment
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 (p. 147), or one destination when it is set up as a single-pipeline channel.	`s3ssl://DOC-EXAMPLE-BUCKET1/sports-thumbnails/delivery/curling`
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 "Reference: identifiers for variable data" (p. 538).	`$dft$`

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" (p. 56).

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 (p. 113).

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 “Step 6: Set up video” (p. 204).

Creating an HLS output group

When you planned the workflow for your channel (p. 73), 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” (p. 169)
     - the section called “Destination fields – MediaStore” (p. 173)
     - the section called “Destination fields – MediaPackage” (p. 177)
     - the section called “Destination fields – HTTP server” (p. 181)
   - **HLS settings** – This section contains fields for the destination of the outputs (p. 181), for resiliency (p. 190), and for captions (p. 191).
   - **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 (p. 422).

- **Manifest and segments** – This section contains fields for configuring redundant manifests (p. 189), for configuring the manifest contents (p. 190), and for configuring media segments (p. 190).
- **DRM** – This section contains fields for configuring encryption of outputs (p. 191).
- **Ad marker** – This section contains fields for setting up for SCTE-35 ad avails (p. 191).
- **Captions** – This section contains fields for configuring captions (p. 191).
- **ID3** – This section contains fields for setting up for ID3 (p. 192).

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" (p. 169)
     - the section called "Destination fields – MediaStore" (p. 173)
     - the section called "Destination fields – MediaPackage" (p. 177)
     - the section called "Destination fields – HTTP server" (p. 181)

     This section also contains fields for the HLS container (p. 188).
   - **Stream settings** – This section contains fields for the output streams (p. 189) (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 "Step 9: Save channel" (p. 210).

Fields for the output destination – sending to Amazon S3

When you planned the destinations for the HLS output group (p. 113), 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

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" (p. 172).

To design the path

1. Collect the bucket names that you previously obtained (p. 113) 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

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 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 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 *curling*: `s3ssl://DOC-EXAMPLE-BUCKET/sports/delivery/curling.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-high.m3u8`
Media files (segments)	`protocol bucket path baseFilename nameModifier optionalSegmentModifier counter extension`	The path for the file for the 230th segment might be: `s3ssl://DOC-EXAMPLE-BUCKET/sports/delivery/curling-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-high.m3u8`

  `s3ssl://DOC-EXAMPLE-BUCKET1/sports/delivery/curling-high.m3u8`

  or

  `s3ssl://DOC-EXAMPLE-BUCKET/sports/delivery/curling-high.m3u8`

  `s3ssl://DOC-EXAMPLE-BUCKET1/sports/redundant/curling-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-high.m3u8`

  `s3ssl://DOC-EXAMPLE-BUCKET/sports/redundant/curling-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 (p. 538).

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 curling-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 curling-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://
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.

Portion of the destination path	Location of the Field	Description
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" (p. 56).
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 (p. 114).

Fields for the output destination – sending to MediaStore

When you planned the destinations for the HLS output group (p. 114), 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" (p. 176).

To design the path

1. Collect the data endpoint for the container or containers. You previously obtained (p. 114) 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 *curling*: `mediastoressl://a23f.data.mediastore.us-west-2.amazonaws.com/delivery/curling.m3u8`
Child manifest files	`protocol dataEndpoint path baseFilename nameModifier extension`	The path for the child manifest for the high-resolution renditions of the curling output `mediastoressl://a23f.data.mediastore.us-west-2.amazonaws.com/delivery/curling-high.m3u8`
Media files (segments)	`protocol dataEndpoint path baseFilename nameModifier optionalSegmentModifier counter extension`	The path for the file for the 230th segment might be: `mediastoressl://a23f.data.mediastore.us-west-2.amazonaws.com/delivery/curling-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/
  curling.m3u8
  ```

  ```
  mediastoressl://fe30.data.mediastore.us-west-2.amazonaws.com/delivery/
  curling.m3u8
  ```

  or

  ```
  mediastoressl://a23f.data.mediastore.us-west-2.amazonaws.com/delivery/
  curling.m3u8
  ```

  ```
  mediastoressl://fe30.data.mediastore.us-west-2.amazonaws.com/redundant/
  curling.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/
  curling.m3u8
  ```

  ```
  mediastoressl://a23f.data.mediastore.us-west-2.amazonaws.com/redundant/
  curling.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 (p. 538).

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 `curling-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 `curling-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.

Portion of the destination path	Location of the Field	Description
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" (p. 56).
3. In the **CDN** settings section, choose `Hls media store`.
4. If the MediaStore user gave you values to configure the connection (p. 116), enter those values in the fields in the **CDN** settings section.

Fields for the output destination – sending to MediaPackage

When you planned the destinations for the HLS output group (p. 115), you might have decided to send the output to MediaPackage. 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" (p. 179).

To design the path

1. Collect the information you previously obtained (p. 115) from the MediaPackage user:

   - The two URLs (input endpoints is the MediaPackage terminology) for the channel. The two URLs for a channel look like this:

     ```
     6d2c.mediapackage.uswest-2.amazonaws.com/in/v2/9dj8/9dj8/channel
     ```

     ```
     6d2c.mediapackage.uswest-2.amazonaws.com/in/v2/9dj8/e333/channel
     ```

     where:

     `6d2c.mediapackage.uswest-2.amazonaws.com` is the domain

     `/in/v2/9dj8/9dj8/` is the folders

     `channel` is the base file name for all the files for this destination. Note that with MediaPackage, the base file name is always *channel*, for every output.

     The two URLs are always identical except for the folder just before `channel`.

Make sure that you obtain the URLs, not the channel name.

- The user name and password. MediaPackage always requires user authentication.

2. You must design the portions of the destination paths that follow the URLs.

Topics

- The syntax for the paths for the outputs (p. 178)
- Designing the nameModifier (p. 179)
- Designing the segmentModifier (p. 179)

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 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: `https://6d2c.mediapackage.uswest-2.a in/v2/9dj8/9dj8/ channel.m3u8`
Child manifest files	`protocol channelURL nameModifier extension`	The path for the child manifest for the high-resolution renditions of the curling output `https://6d2c.mediapackage.uswest-2.a in/v2/9dj8/9dj8/channel-high.m3u8`
Media files (segments)	`protocol channelURL nameModifier optionalSegmentModifier counter extension`	The path for the file for the 230th segment might be: `https://6d2c.mediapackage.uswest-2.a in/v2/9dj8/9dj8/channel-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.
- You must determine 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

- 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 (p. 538).

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:

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 `curling-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 `curling-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	Always `https://`
domain	**URL** in **HLS group destinations** section	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	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. Enter the input user name. 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" (p. 38).
3. In the **CDN** settings section, choose `Hls webdav`.
4. If the downstream system gave you values to configure the connection (p. 116), enter those values in the fields in the **CDN** settings section.

MediaPackage example

This example shows how to set up the destination fields if the downstream system for the HLS ouptput group is 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	**`https://62e3c93793c034c.mediapackage.us-west-2.amazonaws.com/in/v1/9378dje8/channel`**.
Credentials in **HLS group destination A** section	MediaPackage accepts only authenticated requests, so you must enter a user name and a

Field	Value
	password that is known to MediaPackage. 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" (p. 38).
URL in **HLS group destination B** section	**`https://60dei849783734c.mediapackage.us-west-2.amazonaws.com/in/v1/6da5ba717b357a/channel`**.
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 does not 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.

Fields for the output destination – sending to an HTTP server

When you planned the destinations for the HLS output group (p. 116), 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" (p. 184).

To design the path

1. Collect the information that you previously obtained (p. 116) 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" (p. 426), then return to this section.
   - If you are implementing custom paths for manifests, see the section called "Manifests – custom HLS manifest paths" (p. 422), 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 *curling*: `http://203.0.113.55/sports/delivery/curling.m3u8`

File	Syntax of the path	Example
Child manifest files	protocol domain path baseFilename nameModifier extension	The URL for the child manifest for the high-resolution renditions of the curling output `http://203.0.113.55/sports/delivery/curling-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-high-00230.ts`

These destination paths are constructed as follows:

- The operator at the downstream system should have provided you (p. 116) 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 (p. 189), 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" (p. 189).

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 (p. 538).

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 `curling-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 `curling-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 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 URL	Location of the Field	Example
protocol	**URL** in **HLS group destinations** section	**`http://`**
domain	**URL** in **HLS group destinations** section	**`203.0.113.55`**
path	**URL** in **HLS group destinations** section	**`/sports/delivery/`** Always terminate with a slash

Portion of the destination URL	Location of the Field	Example
baseFilename	**URL** in **HLS group destinations** section	**`curling`** 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" (p. 38).
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 (p. 116), 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`**
Credentials in **HLS group destination A** section	If the downstream system requires authenticated 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 information, see the section called "AWS Systems Manager parameter store" (p. 38).
URL in **HLS group destination B** section	For example:

Field	Value
	https://203.0.113.82/sports/curling
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.
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: `curling.m3u8`
- One child manifest for each output: `curling-high.m3u8`, `curling-medium.m3u8`, `curling-low.m3u8`
- TS files for each output:
  - `curling-high-00001.ts`, `curling-high-00002.ts`, `curling-high-00003.ts`, and so on
  - `curling-medium-00001.ts`, `curling-medium-00002.ts`, `curling-medium-00003.ts`, and so on
  - `curling-low-00001.ts`, `curling-low-00002.ts`, `curling-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** 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: **http://p-ep50002.i.akamaientrypoint.net/50002/curling**

Field	Value
	Mapping this URL to the Akamai terminology: • *p-ep* stands for primary entry point *http://p-ep50002.i.akamaientrypoint.ne*t is the hostname • *5002* (in both places) is the stream ID • *curling* is the event 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 information, see the section called “AWS Systems Manager parameter store” (p. 38).
URL in **HLS group destination B** section	For example: **`http://b-ep50002.i.akamaientrypoint.net/50002/curling`** Mapping this URL to the Akamai terminology: • *b-ep* stands for backup entry point *http://b-ep50002.i.akamaientrypoint.ne*t is the hostname • *5002* (in both places) is the stream ID • *curling* is the event 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: **`curling.m3u8`**
- One child manifest for each output: **`curling-high.m3u8`**, **`curling-medium.m3u8`**, **`curling-low.m3u8`**
- TS files for each output:
  - `curling-high-00001.ts`, `curling-high-00002.ts`, `curling-high-00003.ts`, and so on
  - `curling-medium-00001.ts`, `curling-medium-00002.ts`, `curling-medium-00003.ts`, and so on

- `curling-low-00001.ts`, `curling-low-00002.ts`, `curling-low-00003.ts`, and so on

The files will be published to two Akamai hosts – **`p-ep50002.i.akamaientrypoint.net`** and **`b-ep50002.i.akamaientrypoint.net`** and in a folder called **`5002`** on each host.

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" (p. 190) 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" (p. 341).
- **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" (p. 484).

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” (p. 422).

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” (p. 426).

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 “Step 6: Set up video” (p. 204)
- the section called “Step 7: Set up audio” (p. 206)
- the section called “Step 8: Set up captions” (p. 208)

Fields for other HLS features

Topics

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” (p. 484).
- **Program data time**
- **Program date time period**
- **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 (p. 147):

- 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 (p. 192) 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” (p. 453) and specifically the section called “Enabling decoration – HLS” (p. 462).

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 “HLS manifests (embedded captions)” (p. 375).

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” (p. 389).

Creating a MediaPackage output group

When you planned the workflow for your channel (p. 73), you might have determined that you want to include a MediaPackage output group.

Topics

- Procedure to create a MediaPackage output group (p. 192)
- Streams section (p. 192)
- Result of this procedure (p. 194)

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 (p. 112) 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” (p. 484).
7. Choose the first **Settings** link to view the sections for the first output. The section contains fields for the output streams (p. 189) (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 “Step 9: Save channel” (p. 210).

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 “Step 6: Set up video” (p. 204)
- the section called “Step 7: Set up audio” (p. 206)
- the section called “Step 8: Set up captions” (p. 208)

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 “Step 6: Set up video” (p. 204)
- the section called “Step 7: Set up audio” (p. 206)
- the section called “Step 8: Set up captions” (p. 208)

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 MediaPackage channel. For information about SCTE-35 handling in a MediaPackage output, see the section called “SCTE-35 message processing” (p. 453).

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” (p. 387).

Creating a Microsoft Smooth output group

When you planned the workflow for your channel (p. 73), 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 (p. 196).
- **Microsoft Smooth settings** – This section contains fields for the container (p. 197), the connection to the downstream system (p. 196), and resiliency (p. 198).
- **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 (p. 196) and thecontainer (p. 197).
- **Timecode configuration** – This section contains fields for the timecode (p. 198) in the outputs.
- **Sparse track** – This section contains fields for the container (p. 197).

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 (p. 196), and the container (p. 197).
   - **Stream settings** – This section contains fields for the output streams (p. 198) (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 "Step 9: Save channel" (p. 210).

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 (p. 112) 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” (p. 38).
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 “Step 6: Set up video” (p. 204)
- the section called “Step 7: Set up audio” (p. 206)
- the section called “Step 8: Set up captions” (p. 208)

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” (p. 453) and specifically the section called “Enabling decoration – Microsoft Smooth” (p. 463).

Creating an RTMP output group

When you planned the workflow for your channel (p. 73), 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 (p. 200), for resiliency (p. 201), and for captions (p. 201).
   - **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 (p. 199).
   - **Output settings** – This section contains fields for the connection configuration (p. 200).
   - **Stream settings** – This section contains fields for the output streams (p. 201) (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 “Step 9: Save channel” (p. 210).

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 (p. 119) 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 destination** 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” (p. 147), 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” (p. 38).

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 “Step 6: Set up video” (p. 204)
- the section called “Step 7: Set up audio” (p. 206)
- the section called “Step 8: Set up captions” (p. 208)

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” (p. 361).

Creating a UDP output group

When you planned the workflow for your channel (p. 73), 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 (p. 202).
   - **UDP settings** – This section contains fields for setting up ID3 (p. 203) and for resiliency (p. 203).
   - **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 (p. 202) and the connection to the destination (p. 202).
   - **Stream settings** – This section contains fields for the output streams (p. 203) (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 “Step 9: Save channel” (p. 210).

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 (p. 119) 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 (p. 203), 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 “Step 6: Set up video” (p. 204)
- the section called “Step 7: Set up audio” (p. 206)
- the section called “Step 8: Set up captions” (p. 208)

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" (p. 387).

Step 6: Set up the video encode

In the section called "Step 5: Create output groups" (p. 158), 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 (p. 70).

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" (p. 136).

   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

To set up the video encodes in most types of output

1. On the **Create channel** page, find the output group that you created (p. 158).
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.
5. Complete each field as appropriate. For details about a field, choose the **Info** link next to the field.

   - 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.
   - 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.
   - For information about the **Color space** fields, see the section called "Video – color space" (p. 486).
   - For information about the **Additional encoding setting**s fields, see the section called "Video – enhanced VQ" (p. 499)

- For information about the **Rate control** fields, see the section called "Video – rate control mode" (p. 500). 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 "Timecode configuration" (p. 481).

> **Note**
> The **Width** and **Height** fields (which define the video resolution), the fields in the **Framerate** section, and the **Bitrate** field in the **Rate control** section all affect the output charges for this channel. For more information about charges, see the MediaLive price list.

To set up the video encodes in the Frame Capture types of output

1. On the **Create channel** page, find the output group that you created (p. 158).
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 (p. 210).

Creating a video encode by sharing

You can create one video encode and share it among several outputs. Follow the earlier procedure (p. 204) 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.

To create a video encode by sharing

1. On the **Create channel** page, find the output group that you created (p. 158).
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.

Creating a video encode by cloning

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.

To create a video encode by cloning

1. On the **Create channel** page, find the output group that you created (p. 158).
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.

Step 7: Set up the audio encodes

In the section called “Step 5: Create output groups” (p. 158), 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 (p. 70) 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” (p. 136).

   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

To set up the audio encodes from scratch

1. On the **Create channel** page, find the output group that you created (p. 158).
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 (p. 125). You created this selector (p. 154) 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 (p. 207) 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.

To create an audio encode by sharing

1. On the **Create channel** page, find the output group that you created (p. 158).
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.

To create an audio encode by cloning

1. On the **Create channel** page, find the output group that you created (p. 158).
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.

Step 8: Set up the captions encodes

In the section called “Step 5: Create output groups” (p. 158), 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 (p. 70) 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” (p. 136).

   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 by cloning (p. 210)

Creating an captions encode from scratch

To set up the captions encodes from scratch

1. On the **Create channel** page, find the output group that you created (p. 158).
2. Under that output group, find the output where you want to set up an 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 (p. 125). You created this selector (p. 154) 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: Create captions encodes" (p. 373).

Creating an captions encode by sharing

You can create one captions encode and share it among several outputs. Follow the earlier procedure (p. 209) 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 captions encode is nearly identical to the procedure for sharing a video encode or captions encode.

To create an captions encode by sharing

1. On the **Create channel** page, find the output group that you created (p. 158).
2. Under that output group, find the output where you want to set up an 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 an captions encode is nearly identical to the procedure for cloningg a video encode or captions encode.

To create an captions encode by cloning

1. On the **Create channel** page, find the output group that you created (p. 158).
2. Under that output group, find the output where you want to set up an 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: Create captions encodes” (p. 373).
8. Keep in mind that this cloned encode is a new encode instance. If you change fields, you don't affect the source encode.

Step 9: Save the channel

To save (create) the channel, choose **Create channel** in the navigation pane.

You can save the channel only after you have configured and created everything that you require. As soon as you save the 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**.

Creating a channel from a template or by cloning

A channel contains the details that instruct AWS Elemental MediaLive how to transcode (decode and encode) and package your input into specific outputs.

To create a channel, you provide details about inputs, about one or more output groups and their destinations, about the outputs in each output group, and about the video, audio, and caption encodes in each output.

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 the section called "Creating a channel from scratch" (p. 144).
- **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" (p. 212).
- **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" (p. 212).

The procedures in the following topics show how to create a channel by using a template or by cloning. Before you use the procedures, you should understand how to create a channel from scratch. For more information, see the section called "Creating a channel from scratch" (p. 144).

About templates

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" (p. 212).

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.

About 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 input data. Input data is always left blank.

You can edit the existing fields and complete the empty fields as needed.

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.

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” (p. 211).
2. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
3. In the navigation pane, choose **Channels**.
4. On the **Channels** page, choose **Create channel**.
5. 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” (p. 211).
6. Complete the fields, such as the input fields, that must always be completed. You can also edit other fields as needed.

Creating a channel by cloning

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” (p. 212).)

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. It replicates all the data from the base channel *except* for the input sections, which are always empty.
5. Give the channel a new name and complete the input sections. Change other fields as needed.

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**.
3. On the **Channels** page, choose the channel name (not the radio button).
4. 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.

5. (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.
6. 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 of MediaLive can use the template file on the console.

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" (p. 214).

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 the section called "Creating a channel from scratch" (p. 144).
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 (p. 480)), 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" (p. 479).
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" (p. 445).

For the procedure to change the class, see the section called "Changing an existing channel" (p. 450).

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" (p. 213), *Starting, stopping, and pausing a channel* (p. 312), and the section called "Creating a channel by cloning" (p. 212).)
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 MediaLive devices

A MediaLive *device* is the representation within MediaLive, of an AWS Elemental Link *hardware device* that's connected to MediaLive. Use MediaLive device to manage the AWS Elemental Link hardware device that's connected to MediaLive. You can fine-tune the setup of the AWS Elemental Link hardware, transfer use of the device to another AWS account, or accept a device that is being transferred to you.

About MediaLive devices

The AWS Elemental Link hardware that's associated with the MediaLive device is the upstream system that provides the source to MediaLive. The input type associated with the MediaLive device is an Elemental Link input. AWS Elemental Link can stream content to any MediaLive channel.

To clarify the terminology:

- AWS Elemental Link is a physical hardware device. For documentation for the hardware device, see AWS Elemental Link.
- MediaLive device is the representation of AWS Elemental Link within MediaLive. This section describes how to work with the MediaLive device.
- Elemental Link input is a type of input in MediaLive. For information about working with an Elemental Link input, see the section called "AWS Elemental Link source" (p. 90).

The AWS Elemental Link hardware is connected via the internet to MediaLive in a specific Region in your AWS account. You don't need to configure MediaLive for AWS Elemental Link hardware. Instead, the AWS Elemental Link hardware automatically connects itself to your AWS account, and automatically becomes visible to MediaLive.

You set up to use the content from a specific AWS Elemental Link hardware by creating an input that uses the MediaLive device that corresponds to that hardware device. You can then work with the input in MediaLive as you would work with any input. For more information, see the section called "AWS Elemental Link devices" (p. 349).

Topics

Creating a MediaLive device

You don't create MediaLive devices. Instead, when an operator at the upstream system connects an AWS Elemental Link hardware to the internet and powers it on, AWS Elemental Link automatically connects to MediaLive. On the MediaLive console, the **Devices** entry appears in the navigation pane. The new MediaLive device appears in the **Input devices** section.

Editing or viewing details for a MediaLive device

You can fine-tune the setup of the MediaLive device in the following ways:

- Give the device a friendly name
- Edit the device to throttle the delivery bitrate
- Edit the device to use a specific source

You can also view all the settings for the connection between MediaLive and AWS Elemental Link.

To edit or view an input

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Devices**.
3. In **Devices**, go to the card for the MediaLive device that you want, and choose the hyperlink. If there are many MediaLive devices, enter part of the name to filter the list.
4. The details page for this MediaLive device shows information about the device, including the following:

    - The unique ID of the AWS Elemental Link hardware (displayed at the top of the page).
    - 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 status of the connection (p. 218) to the AWS Elemental Link hardware.
    - A device ARN that includes that unique ID.
5. (Optional) On the **Device** page, choose **Edit**.
6. Change any field as needed:

    - **Name** – Enter a name for the device. This name supplements the generated name (such as hd-re87jr7crey).

      We recommend that you include a prefix in the name—`hd-` to indicate that this is an HD device.
    - **Input source** – This field configures the AWS Elemental Link hardware for which content to send —SDI content or HDMI content. The AWS Elemental Link hardware can receive content at an SDI port and an HDMI port. Typically, in a production environment, the device receives content at only one port. Therefore, you can set this field to **Auto**. In some situations, the operator of the AWS Elemental Link hardware might instruct you to set this field to **SDI** or **HDMI**.
    - **Maximum bitrate** – Enter a bitrate in bits/sec only if you want to throttle the bitrate that AWS Elemental Link uses to deliver the stream. Leave this field blank to let AWS Elemental Link determine the bitrate that is best for the network conditions between the hardware device and MediaLive.
7. Choose **Update**.

    Wait for the input **State** to return to **In use** or **Idle** before performing another action with this input.

The new values appear in **Active input** and **Max bitrate** in the **Device settings**. MediaLive sends the new values for these settings to AWS Elemental Link, so that AWS Elemental Link can update itself.

Transferring a device

The owner of a device can initiate an outgoing transfer to a different AWS account, to transfer ownership of the device to that account. The recipient of a transfer can accept or reject the incoming transfer.

Topics

Initiating a device transfer

You can transfer a device to a different AWS account.

To transfer a device to another AWS account

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Devices**.
3. In **Input devices**, go to the card for the MediaLive device that you want to transfer and choose the link.
4. On the details page for the device, choose **Transfer device**.
5. On the **Transfer input device** dialog box, enter the AWS account to transfer to, and type an optional message.
6. Choose **Transfer**.
7. In the navigation pane, choose **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 **Devices**.
3. In **Input devices**, choose **Device transfers**, then choose the **Outgoing transfers** tab.
4. 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, a distributor of AWS Elemental Link might sell you a device, then transfer the device to your account. Or 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 **Devices**.
3. In **Input devices**, choose **Device transfers**, then choose the **Incoming transfers** tab.
4. In the list of transfers, choose the device that you want to accept, then choose **Accept** or **Reject**.
5. In the navigation pane, choose **Devices** again. The device now appears in the list of devices on the **Input devices** page.

Viewing the status of a MediaLive device

To verify the status of the MediaLive device, view the details (p. 215). On the details page, the **Connection state** and **Device state** below the thumbnail define the status.

- **Disconnected** and **Idle** – The MediaLive device is not connected to the internet.
- **Connected** and **Idle** – The MediaLive device is connected to the internet but it is not sending content.
- **Connected** and **Streaming**– The MediaLive device is connected to the internet and is sending content.

Deleting a MediaLive device

You don't delete MediaLive devices. Instead, if the upstream system deregisters the AWS Elemental Link hardware, the MediaLive device no longer appears in the **Devices** section. Note that this is the only way that the MediaLive device is removed.

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

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 (p. 72)—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 “Reference: Supported input containers and codecs” (p. 529).
- **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 “Reference: Supported input containers and codecs” (p. 529).
- **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 “Reference: Supported input containers and codecs” (p. 529).
- **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” (p. 445).

  For information on the classes applicable to each input type, see the section called “Supported input class” (p. 532).
- **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 inputs can be set up as dynamic inputs.

  For more information, see the section called “Input switching” (p. 402).

Inputs, input security groups, and channels

The input is one of the components of a MediaLive workflow. The others are the input security group (p. 71) and the channel (p. 70). 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

To provide information about the source of the video asset, you must create an AWS Elemental MediaLive input.

Topics

Getting ready

Before you create any input, you should plan the workflow. Read the following sections:

- *Setup: Preparing upstream and downstream* (p. 72) – 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” (p. 445) – 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” (p. 351) – 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.

Creating a CDI push input in Amazon VPC

You create a CDI push input in Amazon Virtual Private Cloud (Amazon VPC) to push uncompressed video content from an upstream system that is on your VPC to MediaLive. Create your input before you create the channel that ingests the input.

For information about the characteristics of CDI sources, see the section called "Reference: Supported input containers and codecs" (p. 529).

A CDI input is always a standard-class input. However, you can attach it to a standard channel or a single-pipeline channel. You should have already read the section called "Pipeline redundancy" (p. 445), to decide the class of channel you want.

> **Note**
> Make sure that the content provider is using the latest version of the AWS CDI SDK on their upstream CDI source device.

To create a CDI push input from Amazon VPC

1. You should have already worked with the Amazon VPC user to set up the VPC for your content (p. 88). Make sure that the Amazon VPC user gives you the following information:

   - 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 **AWS CDI**.
6. 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

7. 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" (p. 222).
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" (p. 479).
9. 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:

   ```
   10.99.39.23:5000
   ```

   ```
   192.0.2.54:5000
   ```
10. 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 CDI 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" (p. 90) in the section about setting up a CDI 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 AWS IAM administrator in your organization about your 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**.

Creating a partner CDI push input in Amazon VPC

A partner CDI input is a specific configuration of a CDI input (p. 221). 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” (p. 384).

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 (p. 221).

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:

```
10.99.39.23:5000
```

```
192.0.2.54:5000
```

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:

   ```
   10.99.39.23:5000
   ```

   ```
   192.0.2.54:5000
   ```
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:

     ```
     10.99.39.23:5001
     ```

     ```
     192.0.2.54:5001
     ```

Creating an Elemental Link input

Create your input before you create the channel that ingests the input.

You create an Elemental Link input by attaching a MediaLive device to the input. This device is the representation within MediaLive of the AWS Elemental Link hardware device.

You can set up an Elemental Link input as single-class or a standard-class input. You should have already read the section called “Pipeline redundancy” (p. 445), to decide the class of channel and class of input you want

To set up a single-class input, you attach one MediaLive device to the input. To set up a standard-class input, you attach two different MediaLive devices to the input.

To create a Link input

1. You should have already arranged with the AWS Elemental Link operator to provide one or two AWS Elemental Link hardware devices (p. 92) with your content. Make sure that the AWS Elemental Link operator gives you the following information:

- The name of the AWS Elemental Link device or devices that are providing the source for the input. For example:

  hd-re87jr7crey

  hd-18zel9mimi
- The AWS Region where the AWS Elemental Link devices are configured to work.

2. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
3. Set the AWS Region to match the Region where the MediaLive device exists.
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 **Elemental Link**.
7. In the **Input devices** section, for **Input class**, choose the class for this input:

   - STANDARD_INPUT
   - SINGLE_INPUT
8. 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.
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" (p. 479).
10. 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.

Creating an HLS pull input

Create your input before you create the channel that ingests the input.

You can set up an HLS source as a single-class or a standard-class input. You should have already read the section called "Pipeline redundancy" (p. 445), to decide the class of channel and class of input you want.

To create an HLS pull input

1. You should have already arranged with the video content provider to set up the upstream system (p. 92) 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 M3U8 manifest from. For example:

  `https://203.0.113.13/sports/curling.m3u8` and

  `https://198.51.100.54/sports/curling.m3u8`
- The user name and password (credentials) to access the upstream server and/or the license server. Keep in mind that if you need credentials for both servers, the credentials must be identical.

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 **HLS**.
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.
8. 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" (p. 38).

   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.
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" (p. 479).
10. 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 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.

Creating a MediaConnect push input

Create your input before you create the channel that ingests the input.

You can set up a MediaConnect source as a single-class or a standard-class input. You should have already read the section called "Pipeline redundancy" (p. 445), to decide the class of channel and class of input you want.

To create an input in MediaLive

1. You should have already worked with the MediaConnect user to request that they set up flows for your content (p. 94). Make sure that the MediaConnect user gives you the following information:

    - The ARN or ARNS for the flows. Make sure that you know which flow is the "first flow" (flow A) and which is the "second flow" (flow B). 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
      ```
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 **MediaConnect**.
6. 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.
7. 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” (p. 228).
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” (p. 479).
9. 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.
10. 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.

Result of these procedures

As a result of this setup, one or two MediaConnect flows exist. Each flow has a source that the upstream system is pushing to. Each flow also has one output for the use of MediaLive. The MediaConnect input specifies the ARNs for those outputs.

For a description of this setup that includes a diagram, see the section called "Result of this procedure" (p. 95) in the section about setting up a MediaConnect source.

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 AWS IAM administrator in your organization about your 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**.

Creating an MP4 pull input

Create your input before you create the channel that ingests the input.

For information about the files that MediaLive supports as MP4 inputs, see the section called “Reference: Supported input containers and codecs” (p. 529).

You can set up an MP4 source as a single-class or a standard-class input. You should have already read the section called “Pipeline redundancy” (p. 445), to decide the class of channel and class of input that you want.

To create an MP4 pull input

1. You should have already arranged with the video content provider to set up the upstream system (p. 96) 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 MP4 files from. For example:

     ```
     s3ssl://DOC-EXAMPLE-BUCKET/filler-videos/main/oceanwaves.mp4
     ```

     ```
     s3ssl://DOC-EXAMPLE-BUCKET/filler-videos/redundant/oceanwaves.mp4
     ```
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 (p. 387). 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**.
5. On the **Inputs** page, choose **Create input**.
6. Complete the **Input details** section:

   - **Input** name – enter a name.
   - **Input type** – choose **MP4**.
7. In the **Input class** section, choose the class for this input:

   - STANDARD_INPUT
   - SINGLE_INPUT
8. 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” (p. 38).
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” (p. 479).

10. 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	emsssl://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

Creating an RTMP pull input

Create your input before you create the channel that ingests the input.

For information about the types of RTMP inputs that MediaLive supports, see the section called “Reference: Supported input containers and codecs” (p. 529).

You can set up an RTMP pull source as a single-class or a standard-class input. You should have already read the section called “Pipeline redundancy” (p. 445), to decide the class of channel and class of input that you want.

To create an RTMP pull input

1. You should have already arranged with the video content provider to set up the upstream system (p. 98) for your content. Make sure that the operator of the upstream system gives you the following information:

    - 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 application name and application instance for the source content. (The application instance is also known as the *stream* or *stream key*.) You need this information to create the RTMP input. For example:

  Application name: `live`, and instance name `curling`

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 (pull)**.
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.

   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” (p. 38).
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” (p. 479).
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:
   - 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.

Creating an RTMP push input

Create your input before you create the channel that ingests the input.

For information about the types of RTMP inputs that MediaLive supports, see the section called “Reference: Supported input containers and codecs” (p. 529).

You can set up an RTMP input as single-class or a standard-class input. You should have already read the section called "Pipeline redundancy" (p. 445), to decide the class of channel and class of input that you want.

To create an RTMP push input

1. You should have already arranged with the video content provider to set up the upstream system (p. 99) for your content. Make sure that the operator of the upstream system gives you the following information:

   - 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 that you attach to the RTMP input. 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`
   - The application name and application instance for the source content. (The application instance is also known as the *stream* or *stream key*.) You need this information to create the RTMP input. For example:

     Application name: `live`, and instance name `curling`
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 *Resources: MediaLive input security groups* (p. 243).
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" (p. 445).
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” (p. 479).
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” (p. 102) 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” (p. 102) in the section about setting up an RTMP push source.

Creating an RTMP push input in Amazon VPC

You create an RTMP push input in Amazon Virtual Private Cloud (Amazon VPC) to push content from an upstream system that is in your VPC to MediaLive. Create your input before you create the channel that ingests the input.

For information about the types of RTMP inputs that MediaLive supports, see the section called “Reference: Supported input containers and codecs” (p. 529).

You can set up an RTMP input as single-class or a standard-class input. You should have already read the section called "Pipeline redundancy" (p. 445), to decide the class of channel and class of input you want.

To create an RTMP VPC push input

1. You should have already worked with the Amazon VPC user to set up the VPC for your content (p. 102). Make sure that the Amazon VPC user gives you the following information:

   - The ID of the VPC.
   - The IDs of the two subnets.
   - The IDs of the security groups for the subnet or subnets.
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" (p. 240).
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" (p. 479).
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" (p. 104) 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 AWS IAM administrator in your organization about your 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**.

Creating an RTP push input

Create your input before you create the channel that ingests the input.

For information about the types of RTP inputs that MediaLive supports, see the section called "Reference: Supported input containers and codecs" (p. 529).

An RTP input is always a standard-class input. However, you can attach it to a standard channel or a single-pipeline channel. You should have already read the section called "Pipeline redundancy" (p. 445), to decide the class of channel you want.

To create an RTP push input

- You should have already arranged with the video content provider to set up the upstream system (p. 104) for your content. Make sure that the operator of the upstream system gives you the following information:

  - 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 that you attach to the RTP input.

    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`

1. 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. 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 *Resources: MediaLive input security groups* (p. 243). 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 *Resources: MediaLive input security groups* (p. 243).
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" (p. 479).
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" (p. 107) 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” (p. 107) in the section about setting up an RTP source.

Creating an RTP push input in Amazon VPC

You create an RTP push input in Amazon Virtual Private Cloud (Amazon VPC) to push content from an upstream system that is on your VPC to MediaLive. Create your input before you create the channel that ingests the input.

For information about the types of RTP inputs that MediaLive supports, see the section called “Reference: Supported input containers and codecs” (p. 529).

An RTP input is always a standard-class input. However, you can attach it to a standard channel or a single-pipeline channel. You should have already read the section called “Pipeline redundancy” (p. 445), to decide the class of channel you want.

To create an RTP VPC push input from Amazon VPC

1. You should have already worked with the Amazon VPC user to set up the VPC for your content (p. 107). Make sure that the Amazon VPC user gives you the following information:

   - 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” (p. 240).
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” (p. 479).
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” (p. 110) 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 AWS IAM administrator in your organization about your 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**.

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” (p. 450).

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**.
3. Choose the name of the input, and then choose **Edit**.
4. 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" (p. 450).
   - 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" (p. 479).
5. Choose **Update**.

   Wait for the input **State** to return to **In use** or **Idle** before performing another action with this input.

Deleting an input

If an input is not attached to a channel, you can delete it, even if it is attached to an input security group.

> **Note**
> If the input is attached to a channel, you can't delete the input. This rule exists to ensure that you don't remove an input from a channel and therefore make the channel unusable.

When you delete an input, the attached input security group (if any) is *not* deleted.

To delete an input

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Inputs**.
3. On the **Inputs** page, find the input that you want to delete, and then look at the **State** column.
4. If the state is **Detached**, then choose **Delete**. If the state is **Attached** and you want to delete both the input and its channel, then delete the channel first. For more information, see the section called "Deleting a channel" (p. 213).

   If the input is an Elemental Link input, MediaLive deletes the input. But the MediaLive 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.

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” (p. 479).
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” (p. 479).

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” (p. 434).

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.

Item	Action	Note
	Delete	The multiplex for this program can be idle or running. The program can't have any associated channel.
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" (p. 436).

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" (p. 436).

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" (p. 440).
2. Edit the channel. For more information, see the section called "Editing a channel" (p. 213).

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" (p. 213).

Working with reservations in AWS Elemental MediaLive

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

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 pay an hourly rate (which is lower than the per-minute rate) for a pool of minutes that your channels consume for a month.

For information on charges for reservations, see the MediaLive price list.

Topics

Input and output reservations

Reservation offerings are available for inputs and for outputs.

Input reservation attributes and matching

An input reservation has these attributes:

- Codec
- Resolution (a range)
- Bit rate (a range)
- Region that the input runs in

An *input reservation* applies to the cost of processing input. For a reservation to apply to an input, the attributes of the input reservation must match the fields in the channel's **Input specification**, and the channel must 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 has these attributes:

- Codec
- Resolution (a range)

- Bit rate (a range)
- Framerate (a range)
- Region that the input runs in

An *output reservation* applies to the cost of processing output. For a reservation to apply to an output, the attributes of the output reservation must match the corresponding fields in the channel configuration, and the channel must 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.

How matching works

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

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

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 framerate of **`29.97 fps`** in the channel configuration falls within the range of a framerate attribute of **`<=30fps`** in the reservation.

For the framerate attribute, there is a match as follows:

- If the channel output framerate is set to a specific framerate, there is a match if the framerate specified in the channel configuration falls within the reservation framerate range. For example, the specified framerate is **`24fps`** and the reservation is **`<=30fps`**.
- If the channel output framerate 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`**.

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

How an input or output reservation is applied

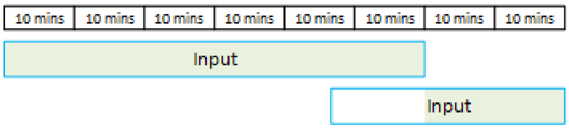
At the start of each monthly billing cycle, AWS replenishes each reservation with the pool of minutes for the month.

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 was running. It accumulates these "running minutes" within the hour, up to a maximum of 60 minutes in the hour.

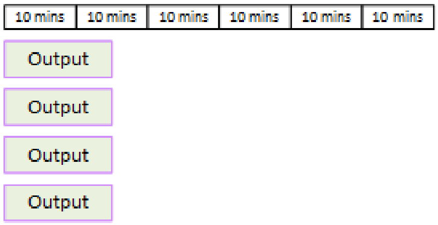
After the reservation minutes are used up for the hour, AWS charges the regular rate-per-minute for the remainder of the items in that hour.

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 green shading in the following illustration.



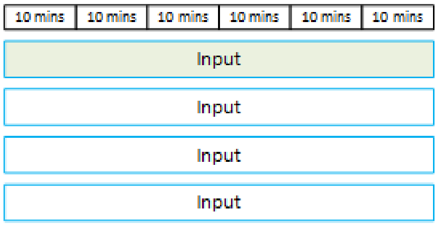
Here is another example of how different items can consume the running minutes. Suppose that in one hour you run only outputs that match a given reservation. You have purchased only one instance of this reservation. You run these four matching outputs simultaneously for 15 minutes each. During that hour, 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 of the minutes in the reservation are not used, 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 for all the inputs or outputs in a given channel to be covered by a reservation.

Add-on reservations

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

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.

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 was running. A channel matches the reservation if the add-on feature is enabled.

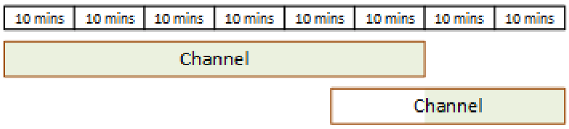
AWS accumulates these running minutes within the hour, up to a maximum of 60 minutes in the hour. After the reservation minutes are used up for the hour, AWS charges the regular rate-per-minute for the remainder for those channels for that hour.

Add-ons are per channel

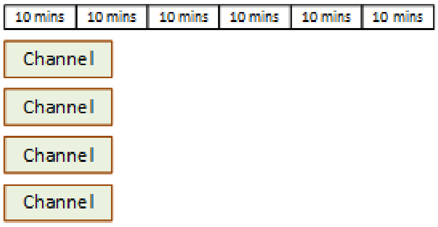
A channel matches the reservation if the add-on feature is enabled one or more times. Within one channel, the number of outputs that use the add-on isn't relevant; the reservation is consumed only once for the entire channel. For example, if there are two outputs in one channel that enable audio normalization,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

Running minutes can be allocated over channels

The same rule applies to add-ons as to input and output reservations (p. 252), 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 green shading in the following illustration.

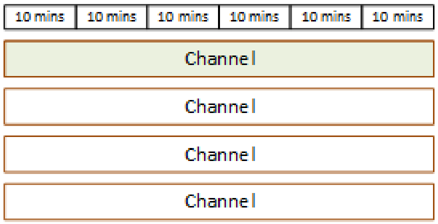


Here is another example of how different channels can consume the running minutes. Suppose that in one hour you run only channels that match the Advanced Audio reservation. You have purchased only one instance of this reservation. You run these four matching outputs simultaneously for 15 minutes each. 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 same rule applies to add-ons as to input and output reservations (p. 252), 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 of the minutes in the reservation are not used, those minutes are lost; the 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**.
3. On the **Offerings** page, complete the **Filter offerings** section to filter for specific offerings. For more information, see Filtering on the Offering Page (p. 255).
4. Choose an offering.

   If you are purchasing a reservation to target a specific output and that output has the framerate set to initialize from source, make sure that you purchase a reservation that specifies "30-60fps". Don't purchase a reservation that specifies "<=30fps".
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.
7. To purchase all the offerings that are displayed on the **Cart** tab, choose **Purchase**.

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 – Framerate (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 using the filters in the left pane:

- You can filter for the reservation type: input, output, or channel (for add-ons).
- You can filter the offerings based on attributes, such as resolution or bit rate.
- **Match existing channel** filters the offerings to show only those offerings that match the inputs and outputs in the chosen channel.
- **Special feature** filters the offerings 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.

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 date in the **End** column.

Deleting a reservation

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

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**.

Working with the 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.

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) into the running channel.

- 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” (p. 258).

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		
Switch the input (perform an input switch)	Yes	Yes	Yes
Prepare the input (perform an input prepare)	Yes	Yes	Yes
Activate a static image overlay	Yes		Yes
Activate a motion graphics overlay	Yes		Yes
Deactivate a 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
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 (p. 257) and start type (p. 257).

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” (p. 402).

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” (p. 395).

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 image overlay action inserts an image overlay and activates it so that it is superimposed on the underlaying video. If the image overlay information includes a duration, then at the appropriate time the image overlay is removed.
- The deactivate image overlay 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.

For information about preparing the image overlay file that the action inserts, see the section called "Image overlays" (p. 392).

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 underlaying 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" (p. 432).

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" (p. 453).

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 and tags actions work

You can set up an action to insert ID3 data in the channel. There are two types of actions:

- Action to insert ID3 metadata in outputs where ID3 passthrough is enabled (p. 388).

  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.
- Action to insert an ID3 tag in each segment in HLS and MediaPackage output packages where ID3 passthrough is enabled (p. 388).

  You specify the ID3 tag as plaintext. MediaLive inserts the tag in every segment.

Before you add ID3 metadata actions to the schedule, read the section called “ID3 metadata” (p. 387).

Before you add ID3 segment tag actions to the schedule, read the section called “ID3 segment tags” (p. 390).

Insert ID3 data 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.

After the channel inserts the message, MediaLive processes the data in the same way as it processes ID3 data that was already in the source content.

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.

After the channel inserts the message, MediaLive processes the data in the same way as it processes ID3 data that was already in the source content.

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 (p. 257) 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" (p. 257).

The general procedure is the same to create any type of action.

To create an action

1. Read the information about planning the actions (p. 258) 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" (p. 284).
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" (p. 404).

Topics

- Fields for a return-to-network message (p. 277)
- Fields for ID3 metadata (p. 278)
- Fields for ID3 segment tags (p. 279)
- Fields for pause (p. 279)
- Fields for unpause (p. 280)

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

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 (p. 148) 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" (p. 404).
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*. For information about these switching rules, see the section called "Fixed, immediate, and follow switches" (p. 404).

Field	Description
	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 (p. 148) 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 (p. 394). 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.
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.

Field	Description
Start type	**Fixed**, **Immediate**, or **Follow**. For information about start types for input prepare, see the section called “Fixed, immediate, and follow switches” (p. 404).
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. For information about these switching rules, see the section called “Fixed, immediate, and follow switches” (p. 404). 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 (p. 148) 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 (p. 387). 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.
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.

Field	Description
	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.
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" (p. 404).
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.

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" (p. 404). 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 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 (p. 148) in this channel.
Start type	**Fixed**, **Immediate**, or **Follow**. For information about start types for input prepare, see the section called "Types of starts" (p. 397).
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 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

Field	Description
	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" (p. 397).
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.
Input attachment	The file input to prepare. The input must already be set up as an input attachment (p. 148) in this channel.
Input clippings settings – Enable input clipping	This field appears only for a file input that is eligible for input clipping (p. 394). 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.

Field	Description
	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.
Start type	**Fixed**, **Immediate**, or **Follow**. For information about start types for input prepare, see the section called "Types of starts" (p. 397).
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. For information about start types for input prepare, see the section called "Types of starts" (p. 397).
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.

Field	Description
Action type	Input Prepare.
Input attachment	The file input to prepare. The input must already be set up as an input attachment (p. 148) 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 (p. 387). 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 (p. 394). 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.

Field	Description
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.
Start type	**Fixed**, **Immediate**, or **Follow**. For information about start types for input prepare, see the section called "Types of starts" (p. 397).
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 an image overlay

This table shows the fields that apply for an action to activate an image overlay.

Field	Description
Action type	Static Image Activate.

Field	Description
Action name	A name for this activation action. For example, 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.
Other fields	Complete these fields to control the behavior of the activation.

Fields for deactivating an 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 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.

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

Field	Description
	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.
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” (p. 38).

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. 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" (p. 404).
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 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" (p. 404).
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.
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" (p. 404).
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**.
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.

Field	Description
Id3	The ID3 metadata. The metadata must be fully formed (including both a header and a frame, as per the ID3 specification) and must be encoded as base64.

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.
Tag	The content of the tag. The content is plaintext. The content can include MediaLive variable data (p. 538). In the following example, the content consists of the date and time, and the current segment number. The tag contents are different in each segment. `$dt$-$sn$`

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” (p. 420) and the section called “Deleting and stopping” (p. 401).
  - 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” (p. 420) and the section called “Deleting and stopping” (p. 401).

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” (p. 281).

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” (p. 284).
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” (p. 284).
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 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” (p. 432).

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.

Window of time for the schedule

The searchable window of time for the schedule extends to approximately one hour in the past. Actions with a start time more than one hour in the past aren't displayed when you view the schedule on the console, and aren't included in the response to a `DescribeChannel` command.

However, a running action that is no longer visible continues to run for its full duration.

The existence of this searchable window of time means the following:

- Actions that relate to input switching: There is no impact.
- Actions that relate to image overlays: For actions to activate a static image overlay for a duration of time, the overlay might still be active, but it is no longer be possible to query MediaLive for the duration.
- Actions that relate to motion graphics overlays: For actions to activate a motion graphics overlay for a duration of time, the overlay might still be active, but it is no longer be possible to query MediaLive for the duration.
- Actions that relate to SCTE-35: For messages that have a duration (splice_insert specified with a duration), the message could still be active, but it is no longer possible to query MediaLive for the duration.
- Actions that relate to ID3 metadata: There is no impact.

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

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.

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

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 (p. 306) 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" (p. 264) and the section called "Deleting actions" (p. 280).
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 AWS Elemental MediaLive API Reference:

```
{
"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 AWS Elemental MediaLive API Reference:

```
{
"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 image action – payload

For information about the meaning and values for the fields in the following JSON, see the AWS Elemental MediaLive API Reference:

```
{
"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 using 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"
          },
          "ImageY": 300,
          "FadeIn": 1500,
          "ImageX": 200,
          "Width": 800,
          "Opacity": 60,
          "Layer": 1,
          "Height": 900
          }
        }
      }
    ]
  }
}
```

Deactivate overlay action – payload

For information about the meaning and values for the fields in the following JSON, see the AWS Elemental MediaLive API Reference:

```
{
 "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 end 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:

```
{
  "ChannelId": "999999",
  "Creates": {
  "ScheduleActions": [
    {
      "ScheduleActionStartSettings": {
        "FixedModeScheduleActionStartSettings": {
          "Time": "2018-05-21T20:42:04.000Z"
        }
      },
      "ActionName": "stop-overlay-029",
      "ScheduleActionSettings": {
        "StaticImageDeactivateSettings": {
          "FadeOut": 500,
          "Layer": 1
          }
        }
      }
    ]
  }
}
```

Activate motion graphic overlay – payload

For information about the meaning and values for the fields in the following JSON, see the AWS Elemental MediaLive API Reference:

```
{
 "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 AWS Elemental MediaLive API Reference:

```
{
"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 the AWS Elemental MediaLive API Reference:

```
{
 "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 AWS Elemental MediaLive API Reference.

```
{
 "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 AWS Elemental MediaLive API Reference.

```
{
 "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 AWS Elemental MediaLive API Reference.

```
{
 "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": "SUQzBAAAAAAAFVRYWFgAAAALAABIZWxsbyBXb3JsZA=="
          }
        }
      }
    ]
  }
}
```

ID3 segment tag item – payload

For information about the meaning and values for the fields in the following JSON, see the AWS Elemental MediaLive API Reference.

```
{
 "ScheduleActions": [
  {
   "ScheduleActionStartSettings": {
    "FixedModeScheduleActionStartSettings": {
     "Time": "string"
    },
    "ImmediateModeScheduleActionStartSettings": {
    }
   },
   "ActionName": "string",
   "ScheduleActionSettings": {
 "HlsId3SegmentTaggingSettings": {
     "Tag": "string"
    }
   }
  }
 ]
}
```

Example

This example of a request creates an ID3 segment tag to be inserted starting at 13:35:59 UTC. The tag contains 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": "$dt$-$sn$"
          }
        }
      }
    ]
  }
}
```

Pause pipeline action – payload

For information about the meaning and values for the fields in the following JSON, see the AWS Elemental MediaLive API Reference.

```
{
 "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” (p. 306).

```
{
  "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 been performed within the last hour

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
        }
      }
    }
  ]
}
```

Using the 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 *Resources: MediaLive input security groups* (p. 243).
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 *Setting up: IAM permissions for production* (p. 21), and specifically the section called “Reference: summary of user access” (p. 50)
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” (p. 311) and the section called “Next steps—experienced video users” (p. 311).

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" (p. 213) 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 *Pricing* (p. 7).

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 *Pricing* (p. 7).

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 "How pause and unpause actions work" (p. 262).

For information about charges for a channel, see *Pricing* (p. 7). 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. Choose **Start**. The channel state changes to one of the following:
   - **Starting**
   - **Running** (encoding on the pipeline or pipelines)
4. Choose the channel name. The details for the channel appear.

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**.

Monitoring a channel or multiplex

You can monitor activity on channels and multiplexes from the AWS Elemental MediaLive console, from Amazon CloudWatch Events, or from Amazon CloudWatch Logs.

Topics

Monitoring a channel using the AWS Elemental MediaLive console

You can view the activity of your channel and its current state.

To monitor activity on a channel (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 the page, see the section called "Editing a channel" (p. 213), *Starting, stopping, and pausing a channel* (p. 312), and the section called "Creating a channel by cloning" (p. 212).)
3. The **Channels** page shows a list of your channels. Each line in the list provides basic information about the channel, including its state:

   - **Creating**
   - **Deleting**
   - **Idle**: The channel isn't running. For information about charges that you accrue when a channel is idle, see *Pricing* (p. 7).
   - **Recovering**: One or both pipelines in the channel failed, but MediaLive is restarting it.
   - **Running**.
   - **Starting**
   - **Stopping**
   - **Updating**: You changed the channel class (p. 386)for a channel.

   Note that Amazon CloudWatch events (p. 327) also capture state changes except a change to updating.
4. To view more details about a channel, choose the name of that channel. The **Channel details** page appears.

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.

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 (p. 503), 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:

   - **Creating**
   - **Deleting**
   - **Idle**: The multiplex isn't running. For information about charges that you accrue when a multiplex is idle, see *Pricing* (p. 7).
   - **Recovering**: One or both pipelines in the multiplex failed, but MediaLive is restarting it.
   - **Running**
   - **Starting**
   - **Stopping**
   -

   Note that Amazon CloudWatch events (p. 327) also capture these state changes.
4. To view more details about a multiplex, choose the name of that multiplex. The **Multiplex details** page appears.

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” (p. 439).

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” (p. 434).

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” (p. 479).

Monitoring activity 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.

To view metrics using the CloudWatch console

Metrics are grouped first by the service namespace, and then by the various dimension combinations within each namespace.

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Metrics**. Under **All metrics**, choose the **AWS/MediaLive** namespace.
3. Choose the metric dimension to view the metrics (for example, choose flow to view metrics per flow).

To view a list of valid metrics stored for your AWS account using the AWS CLI

- At a command prompt, use the following command:

```
aws cloudwatch list-metrics --namespace "AWS/MediaLive"
```

Topics

General information

Collection method

AWS Elemental MediaLive collects raw data at a specified interval (typically every second). After a 10-second window, MediaLive transforms the collected data (for example, it adds the data or averages the data), and reports one datapoint for that window.

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 (p. 351)). 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.

Multiple instances of a metric

You can set up a dashboard on CloudWatch with multiples instances of one metric. For example, one instance that shows a short period, and one that shows a long period.

Global metrics

Active alerts

The total number of alerts that are active.

Details:

- 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

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:

- 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:

- 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 “Timecode configuration” (p. 481).

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:

- 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:

- 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:

- 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” (p. 351).

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:

- 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.

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 (p. 318), 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:

- 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).

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:

- 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.

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:

- 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.

UDP input loss seconds

The number of seconds (the *input loss period*) for which the channel has not received packets from the RTP source. 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:

- 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.

Output metrics

Topics

- Output 5xx errors (p. 327)

Active outputs

The number of outputs that are being produced and successfully written to the destination.

Details:

- 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.

Fill msec

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” (p. 157).

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:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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.

Pipeline locking metrics

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 (p. 442) 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:

- Units: Not applicable.
- Meaning of zero: False (the eligible pipelines are not synchronized), but only if the channel is standard.
- Meaning of no datapoints: The channel is not running.
- Supported dimension sets: ChannelId, Pipeline
- Recommended statistic: Minimum (Value is 0).

Monitoring a channel or multiplex using Amazon CloudWatch Events

MediaLive automatically turns channel or multiplex alert information into events in CloudWatch Events. 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

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 (p. 328), 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 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 “Monitoring using CloudWatch events” (p. 327)) 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 “Monitoring using CloudWatch events” (p. 327)) 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 (p. 331)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 (p. 331)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 (p. 333) 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 `_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 (p. 213) (choose any idle channel), display the **Channel and input details** page (p. 146), 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 (p. 144).

   To enable encoder logs in an existing channel, edit the channel (p. 213); 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" (p. 158).
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 (p. 147) 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 (p. 333).
- `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
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 an IAM 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"
}
```

Maintenance windows

AWS Elemental MediaLive routinely performs maintenance on the underlying systems to ensure their security, reliability, and operational performance. During these maintenance events, channels might need to be restarted.

To view the channels that require maintenance, use the MediaLive console or AWS Health Events in your Personal Health Dashboard.

You can select the day and time that maintenance events will occur. This is called a *maintenance window* and is used every time a maintenance event is required. If you need to change the day and time, the maintenance window can be edited.

You can also manually restart the channels that require maintenance. If you don't set a maintenance window, and don't restart manually, AWS sets the maintenance window. We recommend you set a maintenance window for each channel that requires maintenance.

Topics

Viewing channels that require maintenance

To view the channels that require maintenance, use the AWS Elemental MediaLive console or AWS Health Events in your Personal Health Dashboard.

To view the channels that require maintenance (MediaLive console)

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Channels**.
3. Use the **Maintenance status** column header to sort your channels by **Required by date**.
4. All listed channels must be restarted by the date shown.

To view the channels that require maintenance (Personal Health Dashboard)

1. Open the AWS Personal Health Dashboard at https://phd.aws.amazon.com/phd/home#/.
2. In the navigation pane, choose **Dashboard**, and then select the **Other notifications** tab.
3. Locate notifications with the title **MediaLive maintenance event affecting AWS account 111122223333**. This notification lists the channels that require maintenance, the Region of those channels, and the required by date.

Setting maintenance windows

You can select the day and time that maintenance events will occur. This is called a *maintenance window* and is used every time a maintenance event is required. These windows help to ensure that maintenance has minimal impact to your production. If you need to change the day and time of a maintenance window, you can use the MediaLive console to edit it.

You can also manually restart the channels that require maintenance. If you don't set a maintenance window, and don't restart manually, AWS sets the maintenance window. We recommend you set a maintenance window for each channel that requires maintenance.

To set a maintenance window

1. Open the MediaLive console at https://console.aws.amazon.com/medialive/.
2. In the navigation pane, choose **Channels**. Channels requiring maintenance will display **Required** under the **Maintenance window** column.
3. Select the channel or channels. A unique maintenance window can be set for each channel. Alternately, selecting the upper-most check box in the channel selection column will select all channels.
4. Under the **Channel actions** dropdown menu, select **Edit channel maintenance window**.
5. Select a day and time for maintenance to occur and choose **Save**. Time is presented in UTC.
6. You can verify the window by viewing the **Maintenance window** column on **Channels** dashboard.

The selected day and time is used for all future recurring maintenance events. Repeat these steps to add or edit maintenance windows. After the maintenance is complete, the **Maintenance status** column on the **Channels dashboard** will display **Not required**.

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 the section called "Creating a channel from scratch" (p. 144) and *Resources: MediaLive schedule* (p. 257).

Topics

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 (p. 144), 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 the section called “Creating a channel from scratch” (p. 144).

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:
  - 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” (p. 341).

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.

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 the section called "Creating a channel from scratch" (p. 144).
> 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" (p. 168).

Topics

About audio rendition groups

Standards compliance

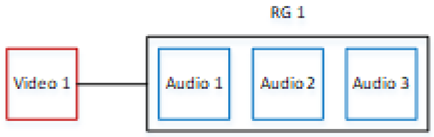
This implementation of audio rendition groups is compliant with *HTTP Live Streaming draft-pantos-http-live-streaming-18* section 4.3.4.1.1.

Examples

Example 1

The HLS output group consists of:

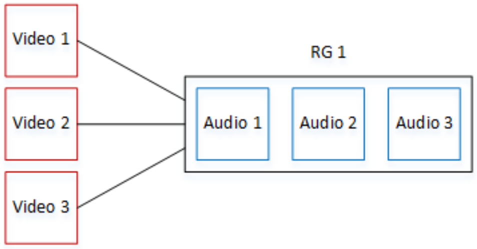
- One video output.
- Three audio outputs (perhaps English, French, Spanish) that all belong to the same audio rendition group.



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.

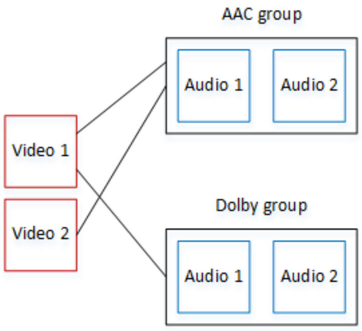


Example 3

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

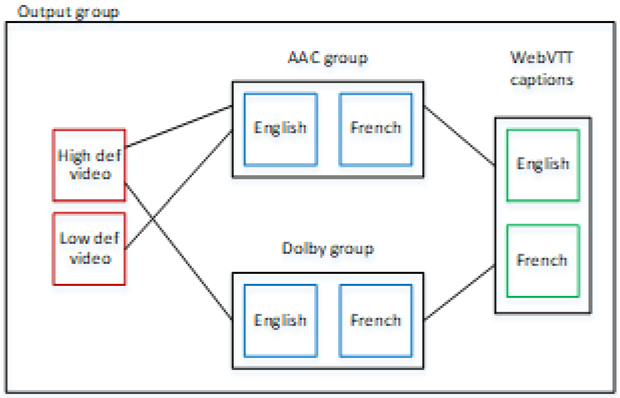
Step 1: Identify the video and audio encodes

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

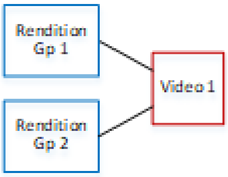
1. Identify any video encodes that you require in the HLS output group. For example, one high-resolution encode and one low-resolution encode.

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 (p. 344) 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.



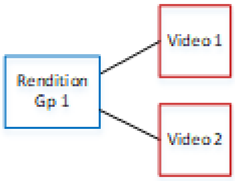
Rules for video and audio in rendition groups

- 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 renditions 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`

Value for a given audio rendition	Client player behavior	Representation in HLS Manifest
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) (p. 189).
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 (p. 345).
   - 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” (p. 347).
- 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 devices

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 device connects to AWS over a secure connection that AWS manages. After the hardware device is connected, it automatically appears in MediaLive as a MediaLive *device*. You then create an *Elemental Link input* that uses this MediaLive device. You can then use the input as you would use any input—you attach the input to a channel.

For information about purchasing an AWS Elemental Link device, see Elemental Appliances and Software.

To clarify the terminology:

- AWS Elemental Link is a physical hardware device.
- MediaLive device is the representation of AWS Elemental Link within MediaLive.
- Elemental Link input is a type of input in MediaLive.

The remainder of this section provides an overview of how you work with the AWS Elemental Link hardware, the MediaLive device, and the Elemental Link input.

Topics

Setting up AWS Elemental Link in MediaLive

Read this section if you have the AWS Elemental Link hardware device in your possession. You might be a regular user of MediaLive who wants to use the hardware device with MediaLive, or you might be someone who is responsible for setting up the device but is not a regular MediaLive user.

To deploy the hardware device

1. To set up the AWS Elemental Link device and connect it to the internet, see the instructions included in the packaging.

   The device is factory-configured to connect to the AWS account and Region that you specified when you purchased the device.
2. Set up permissions in AWS Identity and Access Management (IAM):

   - If you are a regular user of MediaLive, make sure that you or your administrator gives you permission to work with the MediaLive devices. See the section called “Step 1: Requirements for permissions” (p. 26).

- If you are not a regular MediaLive user, you or an administrator in your organization must create an IAM user for you, so that you can use the MediaLive console to manage MediaLive device or devices. See the section called "Creating a non-administrator user " (p. 61).

3. If you need to transfer the device to a different AWS account, follow the steps in the section called "Transferring a device" (p. 216)

Using the AWS Elemental Link device as an input

Read this section if you are a MediaLive user who plans to create a channel that has an input from an AWS Elemental Link device.

MediaLive and AWS Elemental Link work together as follows:

- As soon the hardware device has been set up (as described in the previous section), MediaLive automatically creates a MediaLive device. This device appears on the console (p. 215) in the list of **Devices**.
- To view the MediaLive devices attached to MediaLive in the current Region, see *Resources: MediaLive devices* (p. 215).

  If you don't see a device that you are expecting, it might be in the process of being transferred to your AWS account. You might need to accept the incoming transfer (p. 216).
- You don't need to configure the MediaLive device, except to optionally throttle the delivery bitrate of the AWS Elemental Link device and to specify which of its ports to use.

  For more information on fine-tuning the device, see *Resources: MediaLive devices* (p. 215).
- To use the device in a channel, you create an Elemental Link input in MediaLive, specifying the device as the source. For more information, see the section called "AWS Elemental Link source" (p. 84) and the section called "AWS Elemental Link source" (p. 90).
- When you are ready to use the Elemental Link input in a channel, you attach the input to a channel, in the same way as you would attach any input. For information, see the section called "Input settings—Video selector" (p. 153).

  Typically, you perform this setup after the operator at the upstream system has powered on the AWS Elemental Link 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.

Rules for devices, inputs, and channels

- You can create up to four Elemental Link inputs from each MediaLive device. You can then attach each input to a different channel.

  When you set up inputs in this way, you are not replicating the MediaLive device. There is still only one MediaLive device, but that device is the source for several separate inputs.
- You can attach up to two Elemental Link inputs to one channel. Elemental Link inputs are push inputs, so each one 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 use these two Elemental Link inputs to implement automatic input failover (p. 351).
  - You can include one or both of these Elemental Link inputs in a multiple-input channel, as part of an input switching workflow (p. 402).

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:

- Black video (video failure) – MediaLive will perform a failover if content is considered black for the specified period.
- Audio silence (audio failure) – MediaLive will perform a failover if the specified audio selector is silent for the 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 (p. 445), 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

You can implement automatic input failover 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:

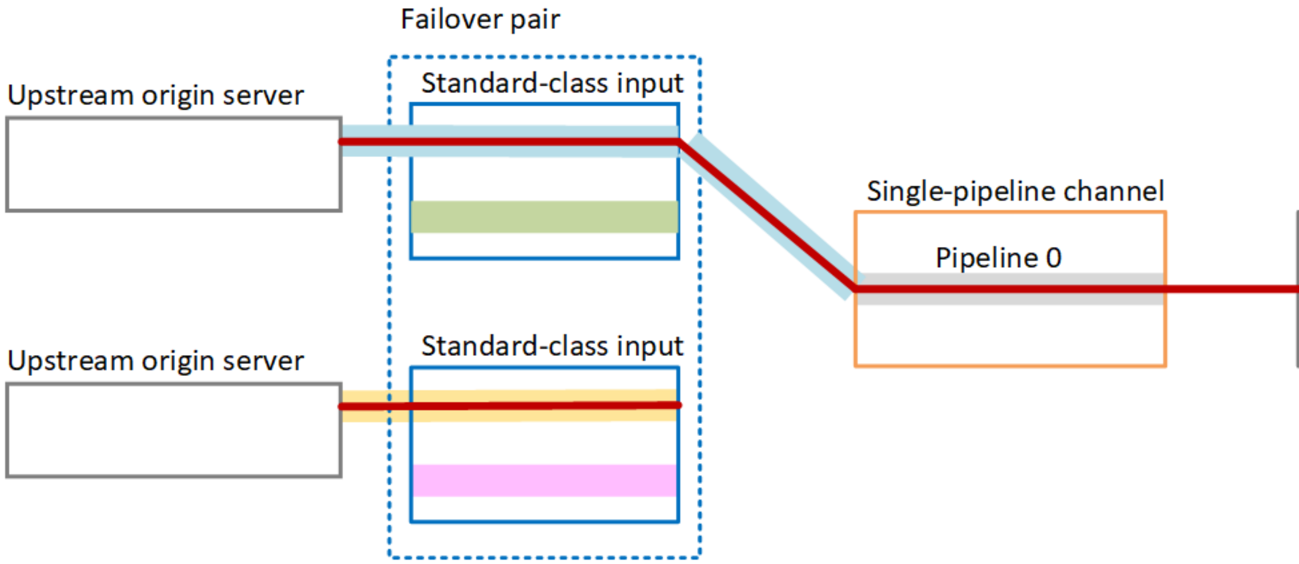
- You can set up two push inputs in the channel, but you won't be able to implement automatic input failover for either of these inputs.
- You can set up one push input in the channel, and you can implement automatic input failover for that input.

Setup

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.

When you start the channel, the channel ingests the content from 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.

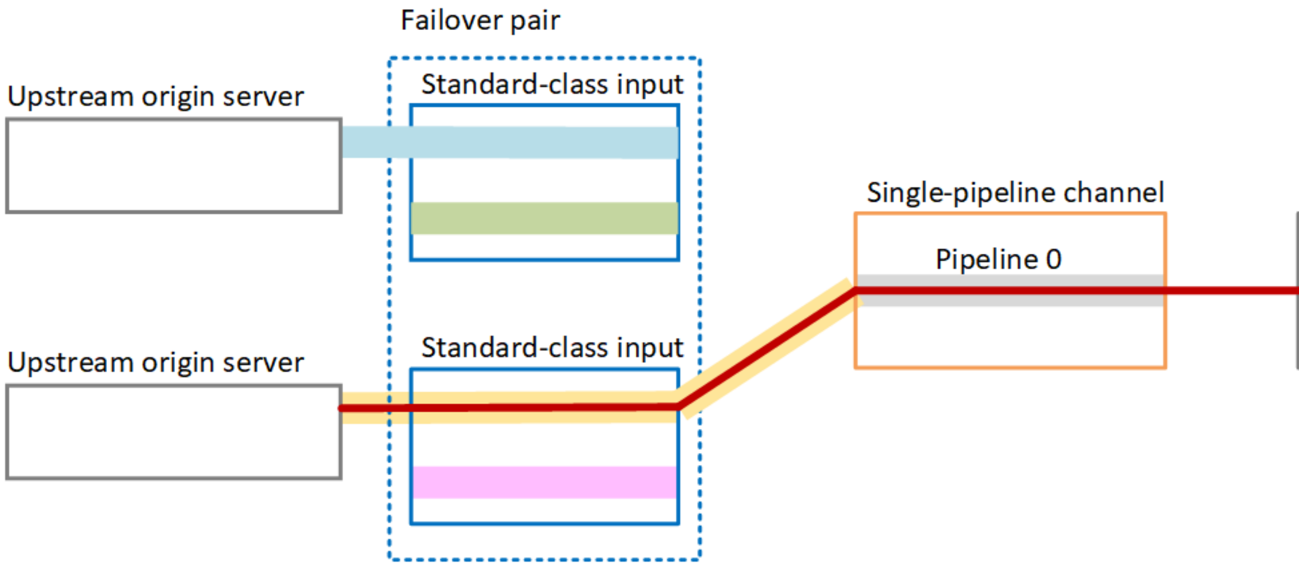


Failure handling

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 single-pipeline channel to protect the channel from failure in the upstream system or the network connection that is upstream of MediaLive.

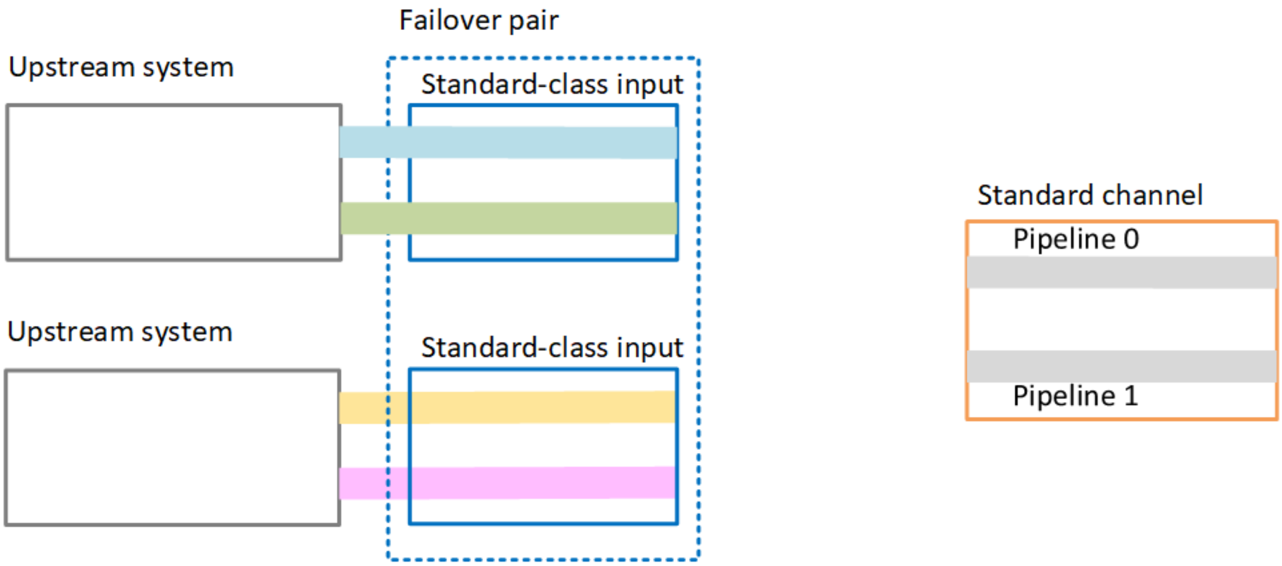
You can implement automatic input failover in push inputs, but not in pull inputs.

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 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.

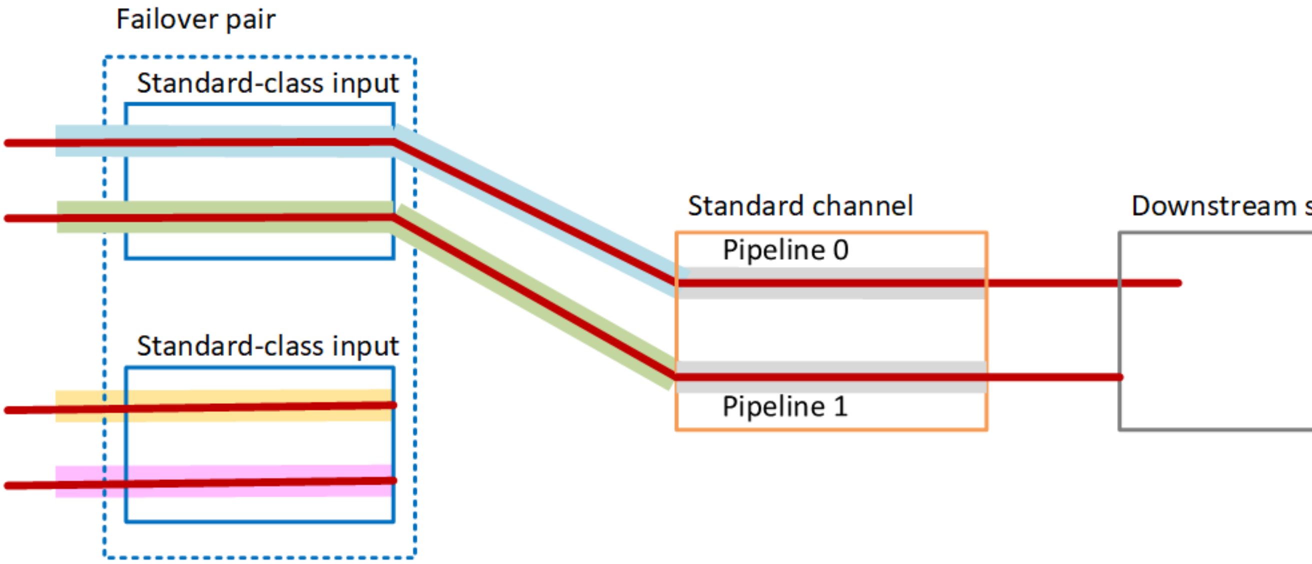
Setup

As this diagram illustrates, there are two instances of the content source. You must make sure that these two instances come from different locations in the upstream system, to provide the desired resiliency in case of a problem upstream of the channel.



When you start the channel, MediaLive ingests the content from both inputs. So it ingests four sources. 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.

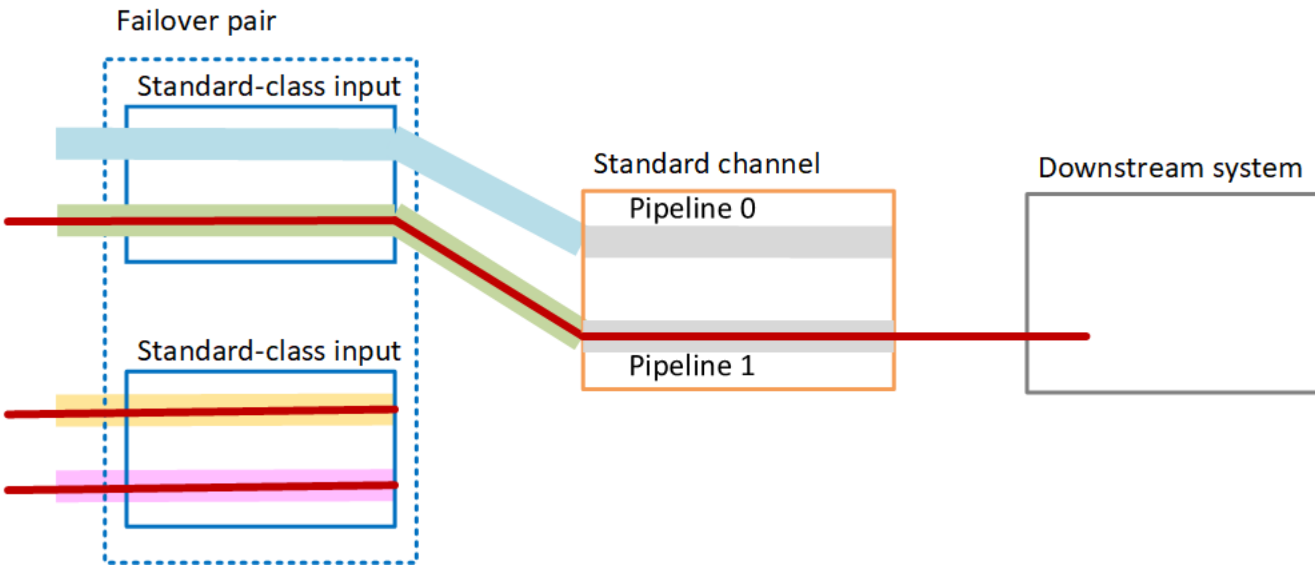
The pipeline produces two outputs for the downstream system, in the usual way. The downstream system chooses to handle one pipeline and to ignore the other pipeline.



Failure handling

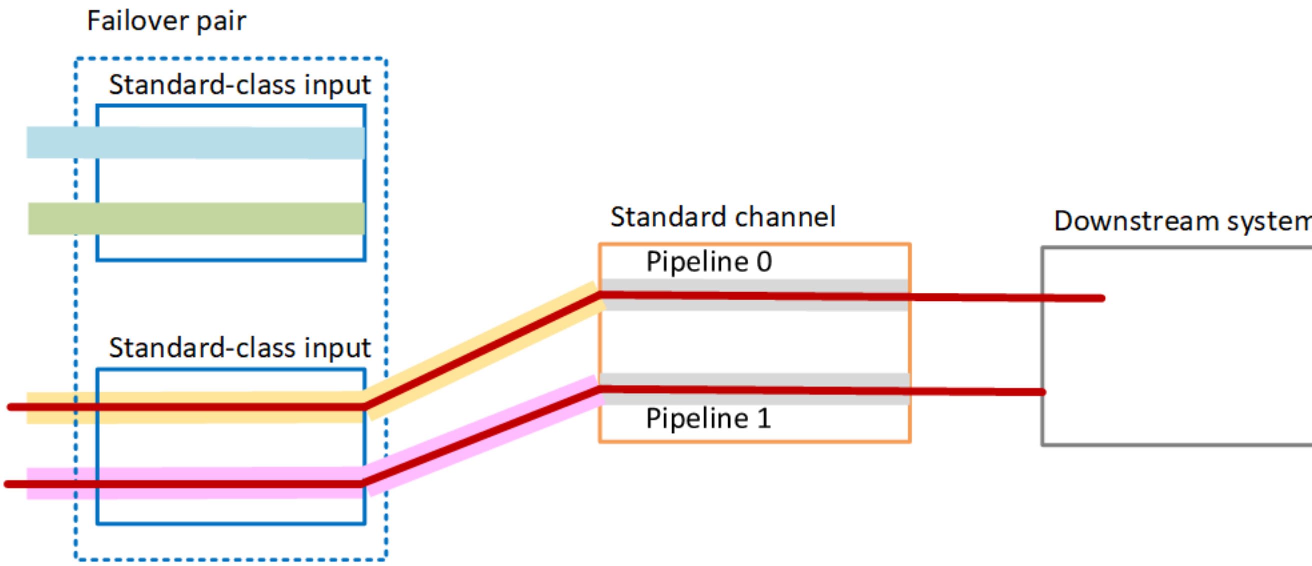
Failure scenario 1 – If normal processing is in progress and there is a failure in pipeline 0, then the recovery behavior for pipeline redundancy occurs:

- The downstream system simply switches over to the other pipeline. For example, it switches from handling pipeline 0 to handling the pipeline 1.
- The downstream system must be able to detect the failure in pipeline 0 and switch over to pipeline 1.



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 (p. 220) and creating a channel (p. 144).

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” (p. 223).

  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 (p. 223).
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 (p. 226) and creating a channel (p. 144).

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 “MediaConnect input” (p. 226), 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” (p. 360).

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 “MediaConnect input” (p. 226), 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 “MediaConnect input” (p. 226) 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.

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 (p. 220) and creating a channel (p. 144).

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 *Resources: MediaLive input* (p. 219) 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” (p. 360).

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” (p. 358) or the section called “Setting up: MediaConnect inputs” (p. 357).
2. To manually switch, create an input switch action (p. 264) 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” (p. 264).

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.

Working with captions

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 captions 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 the section called “Creating a channel from scratch” (p. 144). It also assumes that you have started creating a channel, including associating an input with the channel.

Topics

Supported features

This section provides information on the different features of captions that AWS Elemental MediaLive supports.

Topics

Supported formats

AWS Elemental MediaLive supports specific formats in inputs and specific formats in outputs. See the section called “Supported formats” (p. 514) 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.

Format support by output container

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 “Reference: supported captions” (p. 514).

Constraints for outputs that use OCR conversion

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 (p. 374).
- 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 languages

If the source includes captions in multiple languages, you can include multiple languages in the output as follows:

- **Embedded passthrough** – For any of the embedded source formats, if you specify embedded as the output format, all languages that are in the input are included in the output. You can't remove any of the languages.
- **Embedded In, Other Out** – For any of the embedded source formats, if you are setting up for “embedded in, other out,” you can specify which languages to extract and include in an output.
- **Teletext passthrough** – If you have Teletext source and want Teletext in the output, then all languages (pages) are included in the output. You can't strip out any languages. In fact, the entire Teletext content is included in the output; you can't strip out any of the pages. Furthermore, Teletext passthrough is supported only in TS outputs.
- **Teletext conversion** – If you have Teletext source and want a different format in the output, you can specify the language to extract from the input and the language to include in an output.
- **ARIB passthrough** – For an ARIB source, the only possible output is ARIB. All the languages that are in the input are included in the output. You can't strip out any languages.
- **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

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 (p. 367), on the output side (p. 374), or on both sides.

Font style options

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 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.
Teletext	TTML	You can set up to copy the font information from the source to the output.
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

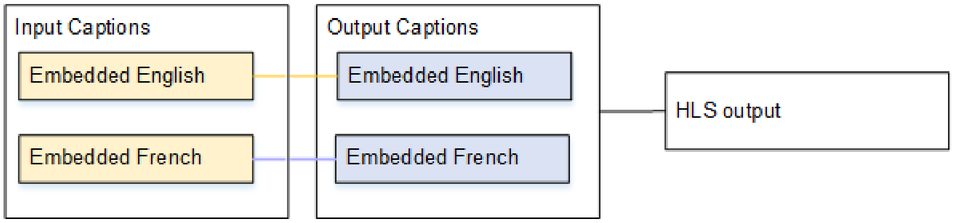
Following are some sample use cases. The use cases are ordered from less to more complicated. They are intended to illustrate many of the capabilities of MediaLive.

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.

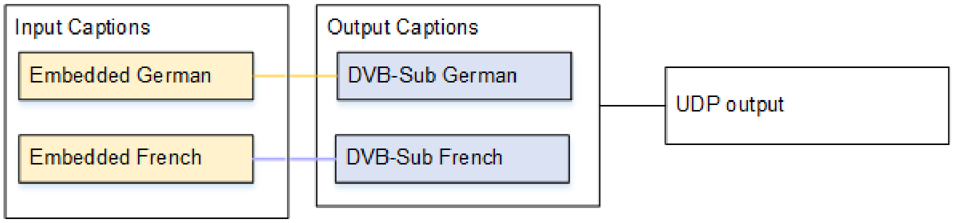
For example, the input has embedded captions in English and French. You want to produce HLS output that includes embedded captions in both English and French.



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.

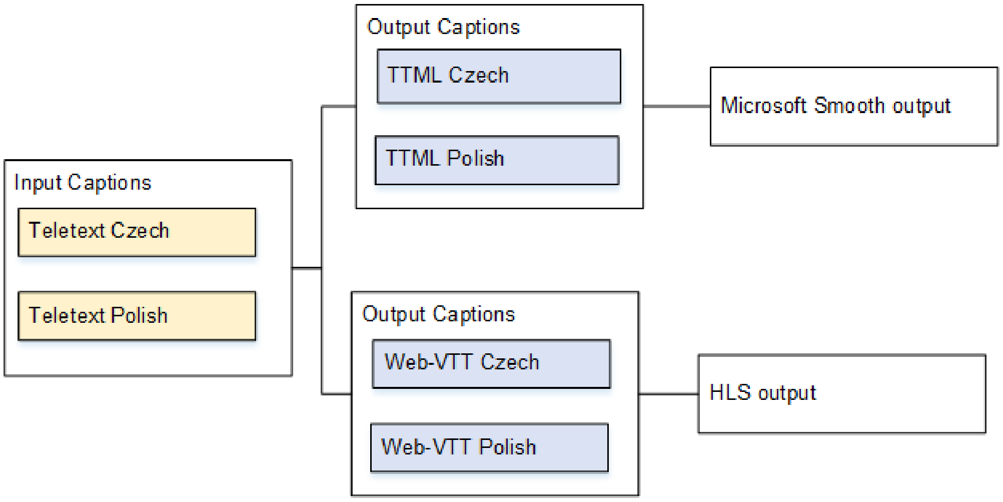
For example, the input has embedded captions in German and French. You want to convert the captions to DVB-Sub and include these captions in both languages in a UDP output.



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.

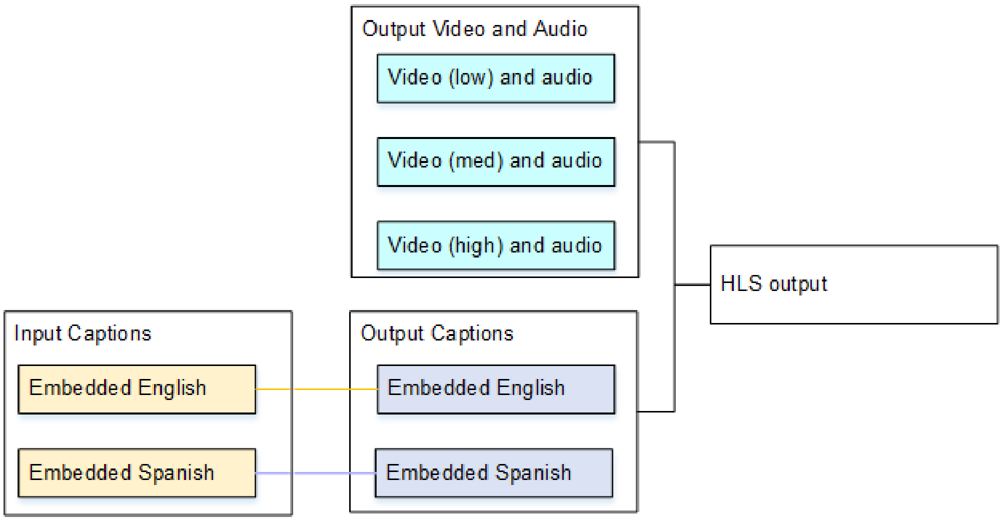
For example, the input has Teletext captions in Czech and Polish. You want to produce a Microsoft Smooth output and an HLS output. In the Microsoft Smooth output, you want to convert both captions to TTML. In the HLS output, you want to convert both captions to WebVTT.



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.



Setting up for captions

When you create a channel, you must specify the format of the input captions, then specify the desired format of the captions for every output. When you save the channel, your choices are validated according to the supported combinations of input container, source captions format, and output container.

Topics

Step 1: Create captions selectors in the input

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.

To 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.

   If you are converting DVB-Sub or SCTE-27 captions to WebVTT, see the section called “Constraints for outputs that use OCR conversion” (p. 362) for limits on the number of languages that MediaLive can ingest.
3. Determine how many captions selectors to create in the input in the channel. Follow this guidance:

- **ARIB passthrough** – Create a single captions selector for all languages (in fact, one captions selector for the entire content). All languages are passed through; there is no other option.

- **Embedded passthrough** – Create a single captions selector for all languages. All languages are passed through; there is no other option. For details, see the section called "Information for embedded " (p. 369).
- **Embedded In, Other Out** – 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 or SCTE-27 In, 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 for outputs that use OCR conversion" (p. 362).
- **Teletext passthrough** – Create a single captions selector for all languages (in fact, one captions selector for the entire content). All languages are passed through; there is no other option. For details, see the section called "Information for Teletext" (p. 370).
- **Teletext conversion** – 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.

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

To create a captions selector

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 selectors**.
3. For **Captions selector name**, enter a suitable name. For example, `Teletext Czech`.
4. For **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 (p. 368), Embedded (p. 369), or Teletext (p. 370).
6. Create more captions selectors, as required.

Information for DVB-Sub or SCTE-27

DVB-Sub and SCTE-27 formats are supported only in RTP inputs. You must provide the following information:

- 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.

PID	Language Code	Result
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-Sub PIDs are included in the output. Not valid for SCTE-27.

- 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.
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

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 outputs

If you followed the instructions in the section called “Step 1: Create captions selectors in the input” (p. 367), 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

1. Identify the types of output media that you plan to create in the channel, for example, Microsoft Smooth and HLS.
2. Identify the combinations of video and audio that you plan to create for each output media.
3. 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 “Reference: supported captions” (p. 514).
4. 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 “Reference: supported captions” (p. 514).
- 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 (p. 363).

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” (p. 517).

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

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 “All captions except sidecar or SMPTE-TT in Microsoft Smooth” (p. 373)
- the section called “Sidecar captions and SMPTE-TT in Microsoft Smooth” (p. 374)

All captions except sidecar or SMPTE-TT in Microsoft Smooth

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.

To set up the output captions

1. In the channel that you are creating, in the navigation pane, find the output group (which you have already created). For example, find the HLS output group.
2. If you have already set up this output group with video and audio, find the outputs where you want to add the captions. Or if you have not set up with video and audio, create a new output in this output group. You set up the captions now, and then set up the video and audio later.
3. Choose the output.
4. For **Stream settings**, choose **Add captions**. You now have an undefined captions encode inside this output.
5. For **Captions description name**, enter a name for this captions asset that is unique in the channel, for example, `Embedded`. Or accept the default (which is automatically generated).
6. For **Captions selector name**, enter the name of the captions selector that you created when you created the captions selectors in the input (p. 367). Specify the selector that identifies the captions asset that is the source for the captions in this output.
7. For **Captions settings**, choose the captions format for the output captions.
8. Complete the fields that appear for the selected format. For details about a field, choose the Info link beside the field. For tips about font styles in DVB-Sub or burn-in, see Font Styles for Burn-in or DVB-Sub Output (p. 374).
9. If the output format is embedded and the output group is HLS, you can include captions language information in the manifest. You perform this setup in the output settings (separate from the captions encode). See HLS manifest (p. 375).
10. If the output format is ARIB or DVB-Sub , you must perform some extra setup in the output settings (separate from the captions encode). See PIDS for ARIB output (p. 374) or PIDs for DVB-Sub output (p. 375).
11. You now have a captions encode that is fully defined.
12. Repeat these steps to create captions in more outputs and output groups, as applicable.

Sidecar captions and SMPTE-TT in Microsoft Smooth

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. See the section called “Captions categories” (p. 517).

You set up each captions asset in its own output within the output group.

To set up the output captions

1. In the channel that you are creating, in the navigation pane, find the output group (which you have already created). For example, find the HLS output group.
2. Create an output in the usual way: in the **HLS outputs** pane, choose **Add output**.
3. Choose the output to show the **Stream settings** pane. The output is set up by default with one undefined video encode and one undefined audio encode.
4. For **Stream settings**, remove the video and audio encodes from this output by choosing the encode and selecting **Remove video** or **Remove audio**. The output is now empty.
5. Choose **Add captions**. You now have an undefined captions encode inside this output.
6. For **Captions description name**, enter a name for this captions asset that is unique in the channel, for example, **`WebVTT Czech`**. Or accept the default (which is automatically generated).
7. For **Captions selector name**, enter the name of the captions selector that you created when you created the captions selectors in the input (p. 367). Specify the selector that identifies the captions asset that is the source for the captions in this output.
8. For **Captions settings**, choose the appropriate format for the output captions.
9. Complete the fields that appear for the selected format. For details about a field, choose the Info link beside the field. For tips about font styles in EBU-TT-D, TTML, and WebVTT captions, see the section called “Font styles for EBU-TT-D” (p. 376), the section called “Font styles for TTML” (p. 376) and the section called “Font styles for WebVTT” (p. 377).
10. You now have one output that contains one captions encode that is fully defined.
11. Repeat these steps to create sidecar captions in this or another output group, as applicable.

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 the output captions are Burn-in or DVB-Sub, you can specify the look of the captions.

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. 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

Complete this step if the output group is UDP/TS and the output captions format is ARIB:

- 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.
Use Configured	Enter a decimal or hexadecimal	This PID is used for the captions.

PIDs for DVB-Sub

Complete this step if the output group is UDP and the output captions format is DVB-Sub.

- 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

Complete this step if the output group is UDP and the output captions format is Teletext:

- 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.

HLS manifests (embedded captions)

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.

- In the HLS output group in Output groups, for **Captions**, 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.
- Still in the HLS output group, for **HLS settings**, in **Captions language mappings**, choose **Add captions language mappings**.
- 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.
- 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 the source captions are embedded or Teletext captions, and the output captions are EBU-TT-D, 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.

To specify font 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 the source captions are embedded or Teletext captions, and the output captions are TTML, you can optionally specify some of the font style information.

To specify font 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 includes 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 **monospaceSansSerif**.	Left blank.
Line gap	Set to leave the line gap unfilled.	Set to leave the gap unfilled.

Font styles for WebVTT

If the source captions are embedded or Teletext captions, and the output captions are WebVTT, you can optionally pass through some of the style information.

No_Style_Data is selected by default. This value outputs only text and timestamp information on the caption encode.

If **Passthrough** is selected, position and color style data will be passed through to the output. This also includes the text and timestamp information.

To specify style information

1. In the output that has the WebVTT captions, display the section for the captions.
2. Set **Style control** to **No_Style_Data** or **Passthrough**.

   Note that when either is selected, there are no fields that you can configure. On **Passthrough**, position and color style data will be passed through to the output. On **No_Style_Data**, only the text and timestamp information will be passed through.

Examples

The following examples describe how to implement the use cases from the section called “Typical scenarios” (p. 365).

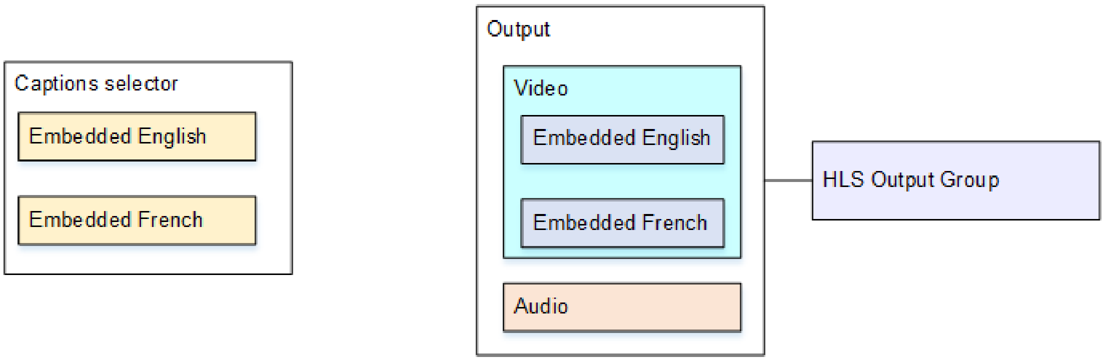
Topics

Use case A: One input format to one output and not converted

This example shows how to implement the first use case (p. 365) 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.

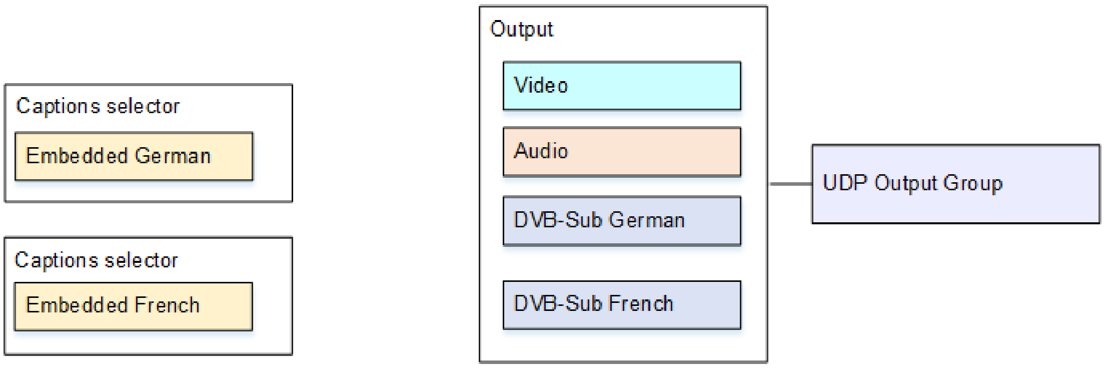


To set up for this use case

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 (p. 365) (p. 365) 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.



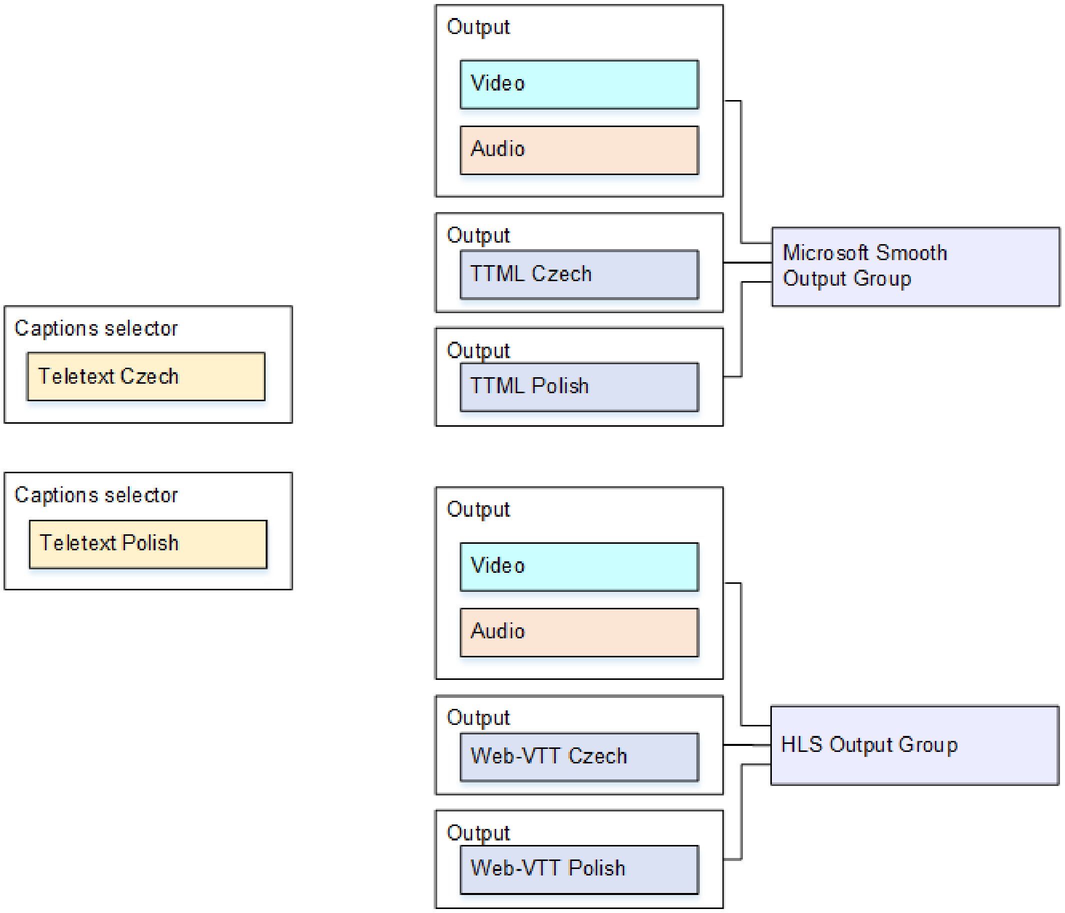
To set up for this use case

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 (p. 365) (p. 366) 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

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.

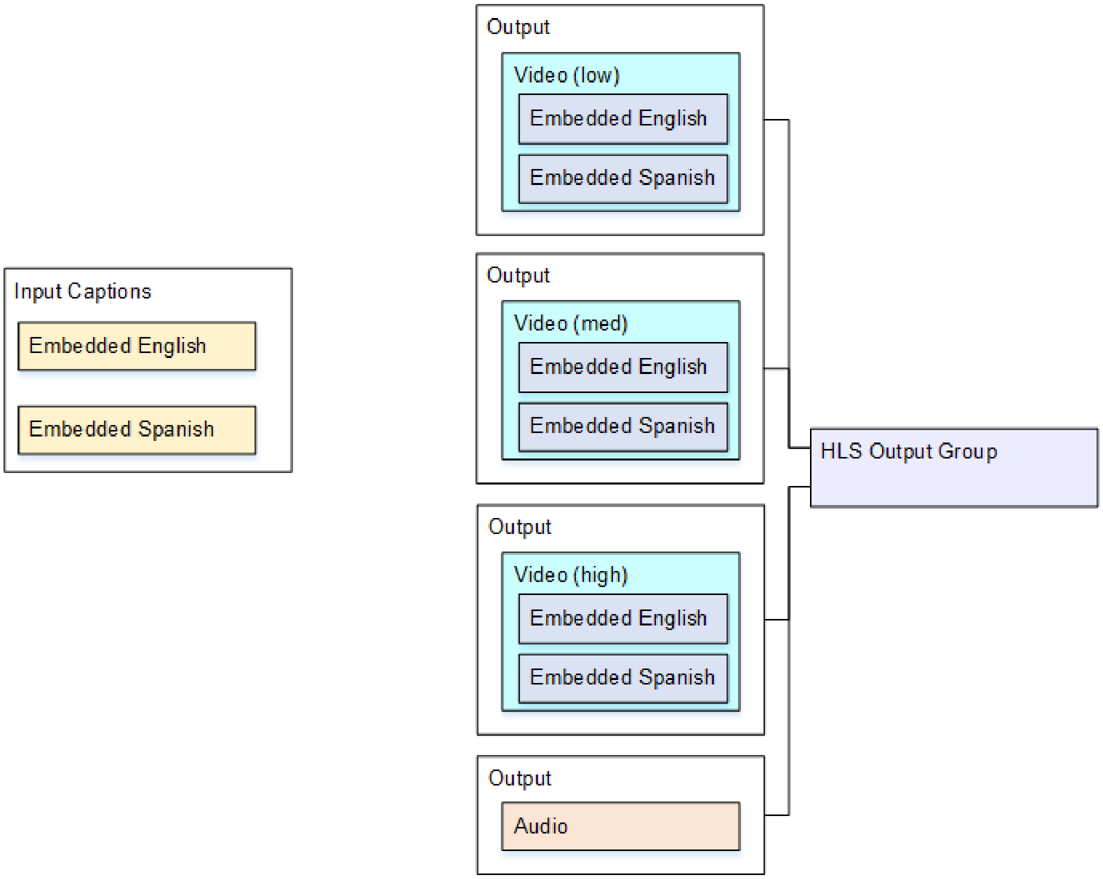
The first setup shows how to set up an ABR workflow when the captions are in the same output as the video, meaning that the captions are either embedded or captions style.

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

This example shows how to implement the fourth use case (p. 365) (p. 366) 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

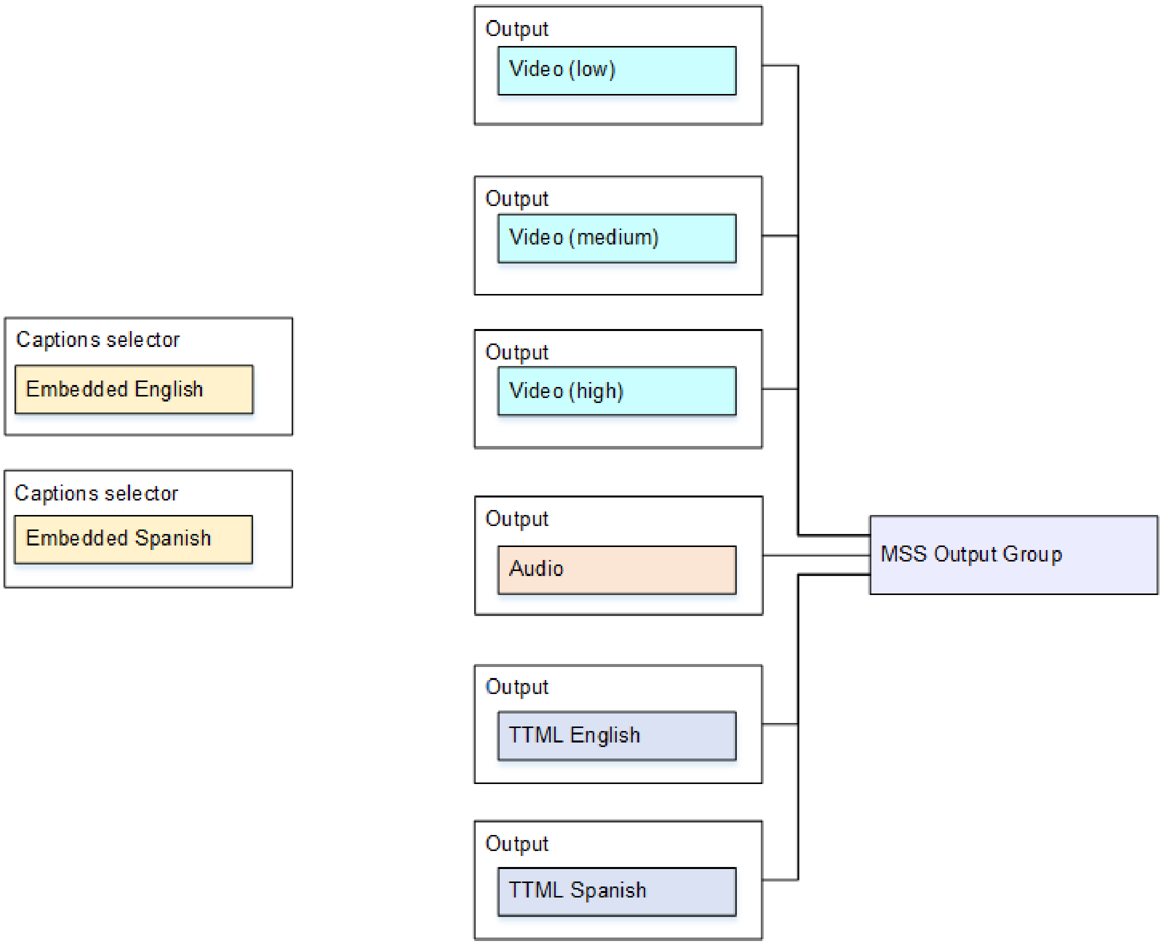
1. In the channel that you are creating, in the navigation pane, in **Input attachments**, choose the input.
2. For **General input settings**, choose **Add captions selector** to create one captions selector. Set **Selector settings** to **Embedded source**.
3. Create an HLS output group.
4. Create 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

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 (p. 351) 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 (p. 221).
	Yes	One set of partner CDI inputs —two CDI inputs set up as partners.

Channel is set up for pipeline redundancy	You want to set up this input for automatic input failover	Type of inputs to create
Yes (standard channel)	No	One regular CDI input (p. 221).
	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” (p. 223).

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” (p. 241).

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 (p. 242) 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" (p. 445).

About channel classes

When you plan the workflow (p. 76), 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" (p. 445).
- 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 (p. 146). You can upgrade or downgrade (p. 450) 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" (p. 445) provides information about choosing the appropriate combination for your workflow.

Channel	Inputs
Standard channel	All inputs must be standard-class inputs.
Single-pipeline channel	The inputs might be a combination of single-class inputs and standard-class inputs.

Channel	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 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" (p. 405).

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 (p. 389) 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" (p. 389) 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" (p. 389) 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" (p. 389) 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” (p. 389) 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” (p. 389).
- 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” (p. 390).

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” (p. 388).

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" (p. 388).
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" (p. 388).
2. Create actions in the schedule. For detailed information, see *Resources: MediaLive schedule* (p. 257).

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

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 tag action in the channel schedule and specify the contents of the tag. At the start time for the action, the channel starts inserting the tag content in every segment in the HLS and MediaPackage outputs. The tag is an ID3 frame of type TXXX.

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.

Comparison to ID3 metadata

A feature that is similar to ID3 segment tagging is ID3 timed metadata (p. 387). 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	ID3 Timed Metadata
One time or repeat?	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 one the ID3 metadata in the applicable outputs, as a one-time event.
Applicable outputs	The applicable outputs are the outputs in the HLS output groups where you have enabled ID3 segment tagging.	The applicable outputs are the outputs in the HLS output groups where you have enabled ID3 metadata.
Clear text or blob?	The tag contents is a string of data in clear text.	The metadata consists of a base64 blob.
Typical contents	Typically, the content consists in all or part of variable text in the form of MediaLive variable data (p. 538). 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.	You are responsible for creating the base64 blob. MediaLive doesn't process the contents in any way. Therefore, the metadata can't contain variable data.

Topics

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 *Resources: MediaLive schedule* (p. 257).

   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 use the static image overlay feature to superimpose a static image onto a video in an MediaLive channel. A static image is a still image that doesn't have motion. You prepare the image and store it outside of MediaLive. You then use the schedule (p. 257) feature in MediaLive to set up a timetable that specifies when images (up to eight different images) will be shown in the running channel, and when each will be hidden.

Examples

Example 1

You want to insert a static image overlay at a specific time and run it for 10 seconds. You want the image overlay to appear in the lower-right corner of the video frame. You want the image overlay to be 50% opaque and to fade in from nothing to full 50% opacity over 2 seconds, then to fade out to nothing starting 2 seconds before the end of the insertion.

Example 2

You want to insert two static image overlays so that they both appear in the video frame either at the same time or with some overlap. You want the display of the image overlays to slightly overlap so that one image overlay appears in a location and, while that image overlay is still showing, another image overlay appears in another location. If the locations overlap either partially or completely, you want to specify which image overlay appears on top.

Features of the static image overlay

The image file must be a 32-bit .bmp, .png, or .tga file, and must not be larger (in pixels) than the input video.

You can configure each image overlay with a start time and duration.

You can insert up to eight images at one time. Each image is a separate "layer." You can set up the overlays to all appear on the underlying video at the same time (or not), and you can set them up to physically overlap each other (or not). You can configure with an opacity and with fade-in and fade-out.

You can insert the image overlay at any position on the video frame, as specified by the **Image X** and **Image Y** fields. For details about a field on the MediaLive console, choose the **Info** link next to the field.

The image is handled as follows:

- The image is overlaid on the underlying video pixel for pixel, without scaling.

  If the overlay is larger than the underlying video or overruns an edge of the underlying video, and if the system can identify this error at channel creation time, you will see an error message at that time.

  If the system can't identify the error in advance, an error message will appear while the channel is running. The channel won't stop, but the overlay request will fail.
- The image is overlaid before creation of individual output encodes (with their different resolutions and video quality). This means that if the underlying video is scaled for a particular output encode, then the image is similarly scaled.
- The image is inserted in all outputs.

Step 1: Prepare the static image overlay file

You must prepare each image overlay that you want to use in your channels. The overlays are stored outside of MediaLive, for example, in an Amazon S3 bucket. An image overlay doesn't belong to MediaLive or to a specific channel in MediaLive. Rather, the image overlays are used by MediaLive.

Follow this procedure to prepare overlays when you need them.

To prepare the overlay file

1. Create a file with the following characteristics:

   - File type: A .bmp, .png, or .tga file.
   - Aspect ratio: The overlay can have any aspect ratio. It doesn't have to match the aspect ratio of the underlying video.
   - Size, in pixels: The overlay can be any resolution (size in pixels) up to the same size as the underlying video.
2. 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. 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.

- Accessible URL that requires SSL. For example:

  ```
  https://203.0.113.0/corporate-logos/large.bmp
  ```
- Accessible URL without SSL. For example:

  ```
  http://203.0.113.254/corporate-logos/high-res.bmp
  ```

4. Make a note of the location. You will need it later.

Step 2: Insert the overlay

You insert an overlay in the video by creating an insert action in the channel schedule. For detailed information, see *Resources: MediaLive schedule* (p. 257) and the section called "Creating actions" (p. 264).

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. So with an image overlay, for example, you create actions in the schedule to specify that a specific image will be overlaid on the underlying video at a specific time, for a specific duration.

When a channel is running, its configuration does not and cannot change. So the channel schedule lets you apply dynamically occurring actions to the channel without having to stop it and reconfigure.

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" (p. 229).
3. Attach the input to the channel in the usual way. See the section called "Step 2: Attach inputs" (p. 148).
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" (p. 264).

   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.

Working with AWS Elemental Link devices

For information on working with AWS Elemental Link devices, see the section called "AWS Elemental Link devices" (p. 349).

For information on working with the Elemental Link input that has an AWS Elemental Link device as its source, see the section called "Elemental Link input" (p. 224).

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 (p. 257). 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" (p. 402).

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” (p. 397).

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” (p. 397).
2. Plan the input switches and input prepares for the channel. See the section called “Planning start” (p. 397).
3. If the associated input switch includes input clipping, see the section called “Input clipping” (p. 394).

   If the associated input switch is an input failover pair, see the section called “Dynamic inputs” (p. 387).
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 *Resources: MediaLive schedule* (p. 257).

Topics

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 (p. 396)to the channel.

1. Stop the channel.
2. Delete (p. 280)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 (p. 148) 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

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 (p. 285).

    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 (p. 285). 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 (p. 387). 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 (p. 394).

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 (p. 351).

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 (p. 399).

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” (p. 281).

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” (p. 280).

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” (p. 264).

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
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.)

- 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 (p. 411) 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” (p. 395).

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* (p. 9).

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* (p. 121) and for creating a channel, as described inthe section called "Creating a channel from scratch" (p. 144).

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* (p. 72).

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

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 (p. 9).
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 “Reference: Supported input containers and codecs” (p. 529).

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 *Resources: MediaLive input* (p. 219). 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” (p. 405).

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 (p. 9).

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” (p. 404) and the section called “Rules and limits” (p. 406).

   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 (p. 416) 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.)

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” (p. 395).

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 *Resources: MediaLive input* (p. 219). 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” (p. 220)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” (p. 387).

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 the section called “Creating a channel from scratch” (p. 144). 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 the section called “Creating a channel from scratch” (p. 144).

Channel and input details pane

On the **Channel and input details** pane for the channel, in the Input specifications (p. 147)section, 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 (p. 418).

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 (p. 412).
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 (p. 416). 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” (p. 417).
- 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 (p. 412):
  - 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” (p. 407).

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” (p. 264).

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” (p. 280).

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.

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 the section called “Creating a channel from scratch” (p. 144).
> 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” (p. 168).

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” (p. 423).

2. Design the paths, paying attention to the syntax and the rules for constructing the paths (p. 182).

   See this guidance for different downstream systems (p. 425).

   See these examples (p. 425).

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 (p. 423). 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 the section called "Creating a channel from scratch" (p. 144).
> 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" (p. 168).

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" (p. 181). 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" (p. 429)
    - the section called "Rules for Akamai" (p. 430)

    These rules supplement the information in the section called "Destination fields – HTTP server" (p. 181).
4. If you also need custom paths for manifests, make sure you read the information in the section called "How custom paths work" (p. 423). 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" (p. 429).

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 (p. 190) 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” (p. 430).

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 baseilenames for the pipelines must be different.
- If you are also implementing custom manifest paths (p. 423), 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 (p. 538) 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 (p. 538) 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 (p. 538) 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 (p. 423). Complete both fields.
Base URL Content A and Base URL Content B	These fields apply only if you are also implementing custom manifest paths (p. 423). 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” (p. 429).

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 (p. 538) that include the date or time.

Field	Rule
	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 (p. 538) 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 (p. 538) 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 (p. 423). Typically, with Akamai CDNs, you do implement 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 (p. 423). 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 (p. 424) and for redundant manifests for your downstream system—either an Akamai CDN (p. 430) or another downstream system (p. 429).

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" (p. 348).

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 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 (p. 257) 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 (p. 433). 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 the section called "Creating a channel from scratch" (p. 144).

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 "Step 4: Complete general settings" (p. 157).

   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 *Resources: MediaLive schedule* (p. 257) and the section called "Creating actions" (p. 264).

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” (p. 438). 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* (p. 8).
- 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” (p. 247).

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 (p. 220).
- 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” (p. 247). 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” (p. 437).
- 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” (p. 437).

For more information about completing the channel fields, see the section called “Creating a channel from scratch” (p. 144).

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.
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). • Highest supported rate is 60 frames per second.

Field	Value
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 “Reference: supported captions” (p. 514).
- 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.
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” (p. 420).

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 *Starting, stopping, and pausing a channel* (p. 312).

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. 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 (p. 387) 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. The output group must be an Archive, HLS, or UDP group. If the output group is HLS, the output must be a standard output (not an audio-only output).

   (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.)
4. For an **Archive** output group, set up the **Output settings** section as follows:
   - Choose **PID settings**.

- For **Nielsen ID3**, choose **PASSTHROUGH**.

5. For an **HLS** output group, set up the **Output settings** section as follows:

   - Choose **Standard hls**.
   - Choose **PID settings**.
   - For **Nielsen ID3 behavior**, choose **PASSTHROUGH**.

6. For a **UDP** output group, set up the **Output settings** section as follows:

   - Choose **Network settings**.
   - Choose **PID settings**.
   - For **Nielsen ID3**, choose **PASSTHROUGH**.

Pipeline locking

Pipeline locking is a feature that controls some of the behavior of the pipeline redundancy (p. 445) feature of MediaLive. When you create a standard channel, in order to implement pipeline redundancy, the channel has two pipelines. The pipeline locking feature ensures that the two outputs are frame-accurate with each other. You can configure the *mode* for pipeline locking. You can't disable pipeline locking.

Topics

Requirements for pipeline locking

Channel requirements

The channel must be a standard channel with only one input attached. It doesn't work in a channel with multiple inputs where your implement input switching.

Output requirements

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 there is no guarantee that the two pipelines will be frame-accurate with each other.

Step 1: Verifying the input

You must speak to your upstream system to make sure that the input source meets these requirements:

- The type of input. The input *cannot* be HLS.
- Whether the input has embedded timecode. These rules apply:
  - For both locking modes (p. 443), 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.
- If the source has varying framerates, each input framerate must be a simple conversion to the desired output framerate. For information about simple conversions, see the requirements later in this section.

If the input doesn't meet these requirements, you can still use it, but there is no point in following the rest of these procedures because MediaLive won't perform pipeline locking in any of the outputs. This means that there is no guarantee that the two pipelines will be frame-accurate with each other.

Frame rate requirements

The conversion between the input frame 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 apply 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 pipeline 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 the section called “Creating a channel from scratch” (p. 144).

Setting the mode and timecode for pipeline locking

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.
4. In the **General settings** section, choose **Timecode configuration**. Set **Source** to **Embedded**. You must set this field, otherwise MediaLive won't attempt to perform pipeline locking.

You can optionally set the **Sync threshold**. This field has no effect on pipeline locking.

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.
2. If necessary, create the outputs and video encodes in each output, then choose the first video encode.
3. In the **Codec settings** field, choose the codec. More fields appear.
4. 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 (p. 442).
5. Continue setting up the channel.

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.
4. If necessary, create the outputs. Then in the **Output settings**, choose **Network settings**. More fields appear.
5. 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.
6. If necessary, create the video encodes in each output, then choose the first video encode.

7. In the **Codec settings** field, choose the codec. More fields appear.
8. 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 (p. 442).
9. Continue setting up the channel.

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.
- 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.
- Make sure that MediaLive supports pipeline locking in the output group that is not synchronized.

- Make sure that you changed the **Framerate control** so that it's 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” (p. 447).

- 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” (p. 448).
- If you don't want to implement pipeline redundancy now or in the future, then see the section called “Single-pipeline channel without upgrade” (p. 449).

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:

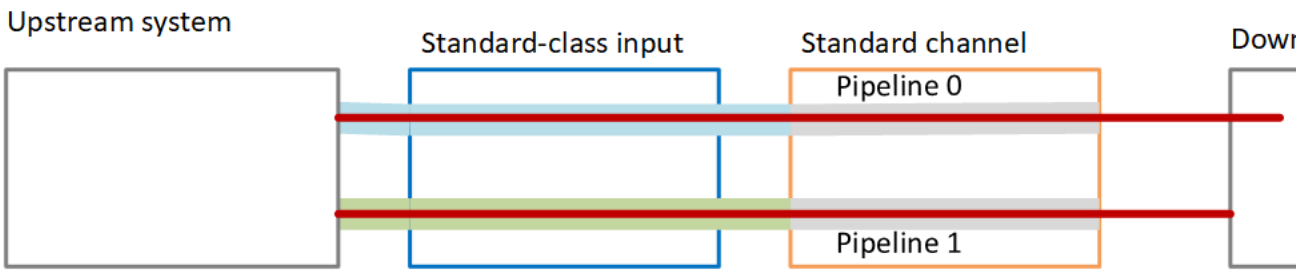
- Make sure that the upstream system can provide you with two instances of the source content. See the section called “Assess source formats and packaging” (p. 78).
- When you get to the step tocreate inputs (p. 87), set up all the inputs as standard-class inputs.

  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**.
- After you have created the inputs and you are at the step where you coordinate with the upstream system (p. 87) about how they will provide the content, make sure that they provide two content sources.
- When you get to the step to create the channel, do the following:
  - Set up the channel as a standard channel. See the section called “Step 1: Complete channel details” (p. 146).
  - At the step to attach inputs to the channel (p. 148), 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.

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).

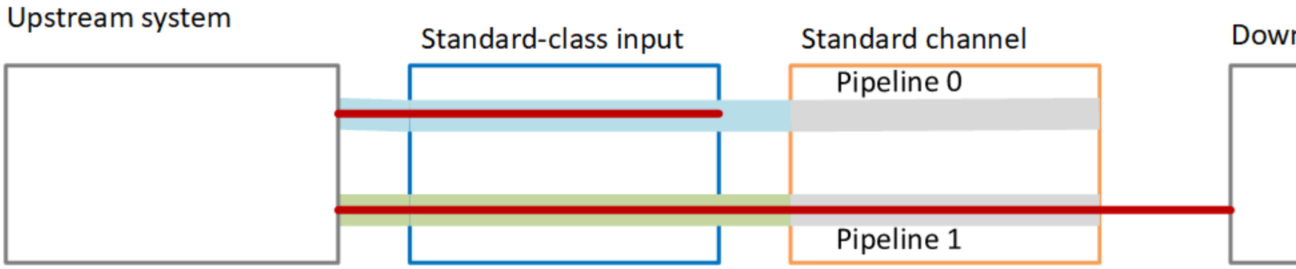


Failure handling

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 get to the step to create inputs (p. 87), set up all the inputs as standard-class inputs.

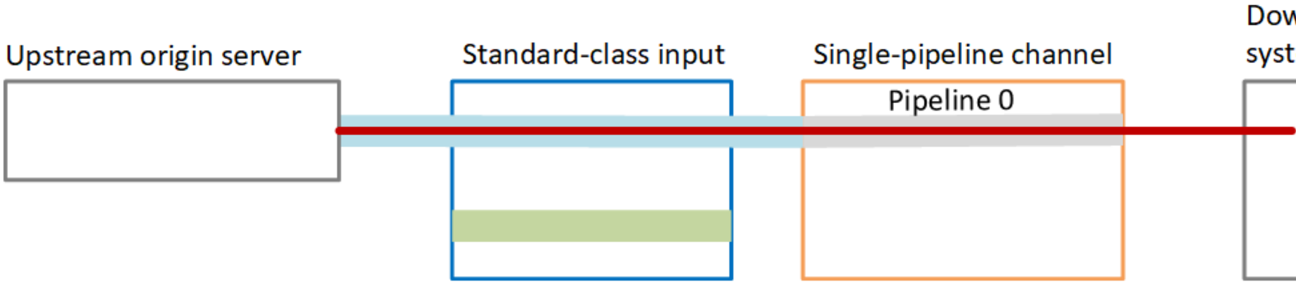
  Some inputs — CDI inputs and RTP inputs — are always set up as standard-class inputs. For all other inputs, set the **Input class** field to **Standard input**.
- After you have created the inputs and you are at the step where you coordinate with the upstream system (p. 87) about how they will provide the content, make sure that they provide *one* content source.
- When you get to the step to create the channel, do the following:
  - Set up the channel as a single-pipeline channel. See the section called “Step 1: Complete channel details” (p. 146).
  - At the step to attach inputs to the channel (p. 148), double-check that the inputs you attach are standard-class inputs.

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 (p. 448).

Follow these guidelines when you plan the workflow:

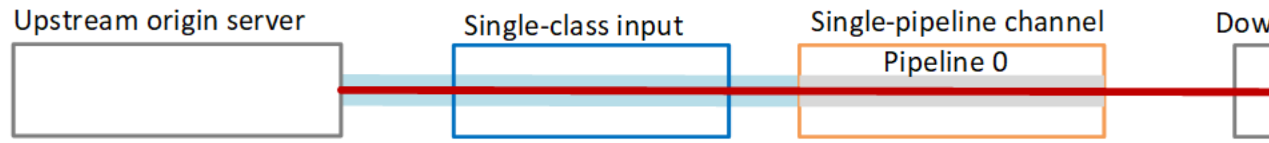
- When you get to the step to create inputs (p. 87), set up the inputs as follows:
  - Set up CDI inputs and RTP inputs as standard inputs, because that's the only way to set them up.
  - Set up all other inputs as single-class inputs. To set up the input in this way, set the **Input class** field to **Single input**.
- After you have created the inputs and you are at the step where you coordinate with the upstream system (p. 87) about how they will provide the content, make sure 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.
- When you get to the step to create the channel, do the following:
  - Set up the channel as a single-pipeline channel. See the section called “Step 1: Complete channel details” (p. 146).
  - At the step to attach inputs to the channel (p. 148), attach the inputs that you have identified. The inputs might be both standard-class inputs and single-class inputs.

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.

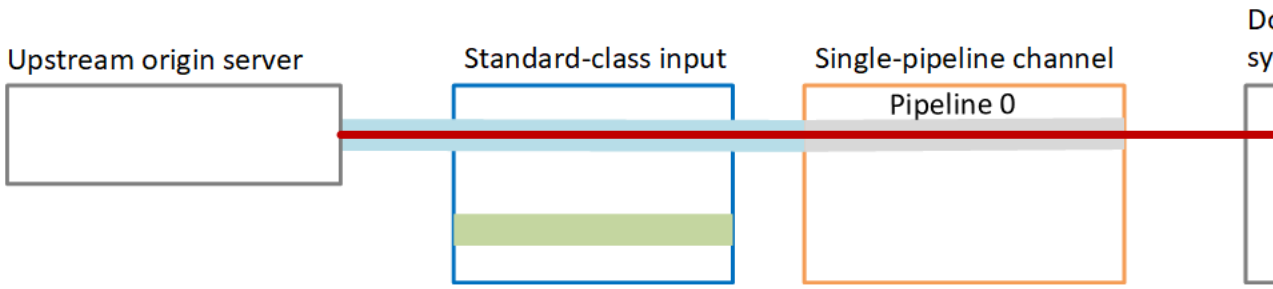
- The channel contains one pipeline—pipeline 0.
- Each single-class input that is attached to the channel contains one pipeline. The input is connected to one content source.

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 (p. 448).

Perform these steps:

- Arrange with your upstream systems to start sending two instances of the source content.
- Stop the channel (p. 312).
- 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 (p. 312).

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 (p. 111) 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 (p. 449).

Perform these steps:

- Arrange with your upstream systems to start sending two instances of the source content.

- Stop the channel (p. 312).
- Detach each single-class input. To detach the inputs, you must edit the channel (p. 213) and remove the attached inputs.
- Edit each input (p. 241) 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 (p. 213) 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 (p. 312).

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 (p. 241) 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 (p. 241) 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” (p. 351).

- Input loss behavior – You can set up the channel to control how MediaLive behaves when 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 “Global configuration - Input loss behavior” (p. 157).
- 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” (p. 445).

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.

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 the section called “Creating a channel from scratch” (p. 144).

Support for SCTE-35 on the input side

On the input side, SCTE-35 messages can appear only in MPEG-2 transport stream (TS) inputs, which means that in MediaLive they can appear only in the following types of inputs:

- Elemental Link inputs
- HLS inputs
- MediaConnect inputs
- RTP inputs

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

About message processing

MediaLive works with the SCTE-35 messages in MPEG-2 transport stream (TS) inputs. These messages might or might not include segmentation descriptors.

Topics

Supported features by input type

The following table shows which inputs might include ad avail information and how MediaLive handles that information.

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		

Supported output features

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.

The behavior that you want must be set up in the channel.

For more information, see the section called “Enabling ad avail blanking” (p. 465) and the section called “Enabling blackout” (p. 469).

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.

The behavior that you want must be set up in the channel. For more information, see the section called “Enabling manifest decoration” (p. 462).

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 “Enabling SCTE-35 passthrough or removal” (p. 472).

Inserting SCTE-35 messages using the schedule

You can insert SCTE-35 messages in TS outputs using the channel schedule (p. 261). 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 SCTE-35 messages using the schedule” (p. 473).

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.

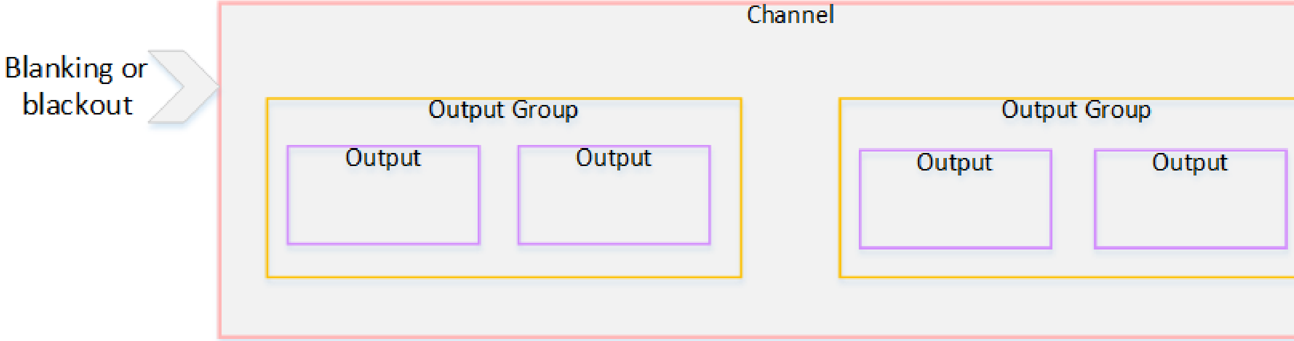
Scope of processing by feature

The SCTE-35 features have different scopes in terms of the output groups and outputs that they affect:

Blackout or ad avail blanking

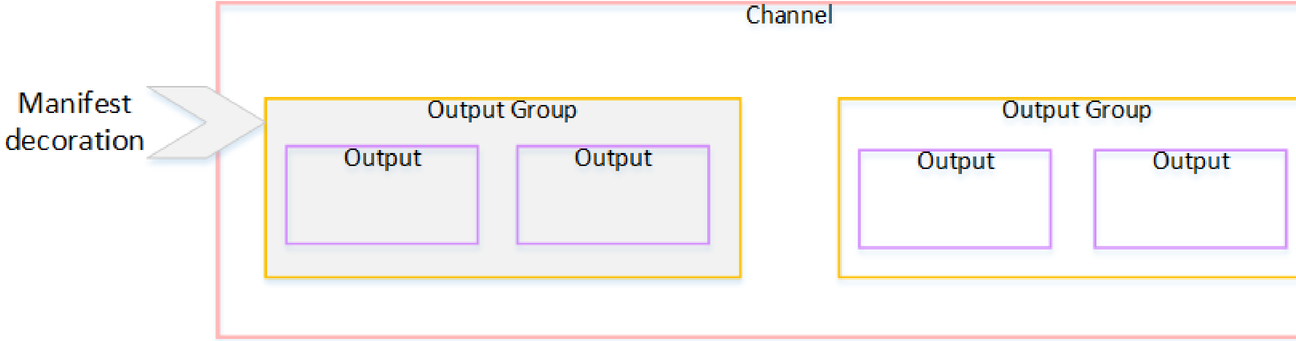
Blackout applies at the *global output* level. If you enable blackout, all the relevant content in every output in every output group is blanked.

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.



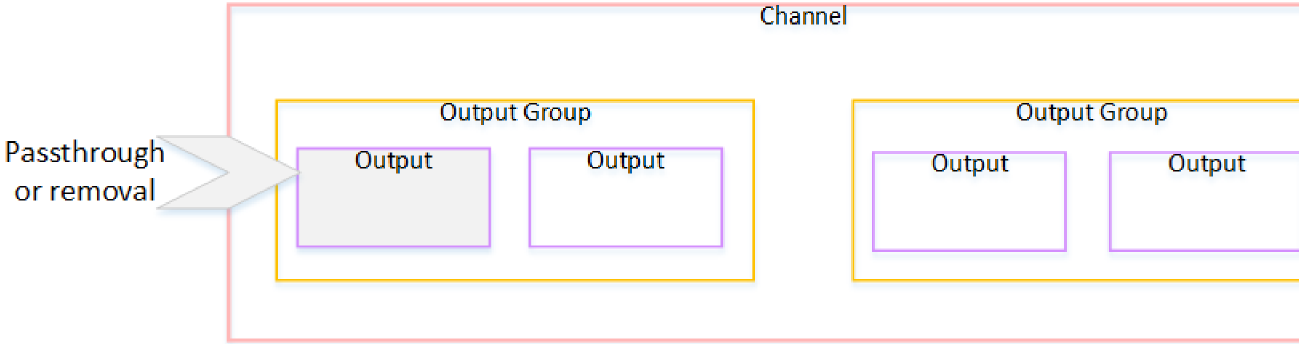
Decoration

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

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 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 TS. This type of source applies to all inputs that can include SCTE 35 messages.
- Tags in an HLS input manifest. This type of source applies only to HLS inputs.

> **Note**
> You only need to follow this 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 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

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 Ad Avail mode to notify MediaLive of the ID type of SCTE-35 messages that the input is using to indicate ad avail events.

To set the ad avail mode

1. In the channel that you are creating, in the navigation pane, choose **General settings**.
2. Choose **Avail configuration**.
3. Set the **Avail settings**:
   - SCTE-35 splice insert (default): Select this mode if the input uses splice inserts to indicate ad avails. The input might also contain messages for others events such as chapters or programs.
   - SCTE-35 time signal apos: Select this mode if the input contains time signals of segmentation type **Placement opportunity**. The input might also contain messages for other events such as chapters or programs.
4. In **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.
5. Leave **web_delivery_allowed_flag** and **no_regional_blackout_flag** as Follow for now. For information about these fields, see the section called “Ad avail blanking restriction flags” (p. 468).

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” (p. 462).

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 (p. 474).
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" (p. 462).
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" (p. 462)). SCTE-35 messages that are *not* ad avails result in the insertion of blackout start/end instructions, assuming that blackout is enabled. If blackout is not enabled, these instructions are not inserted.

Splice insert mode

Message type ID: splice insert

Segmentation Type ID	Base64	Cue-out, Cue-in	Blackout
No segmentation descriptor present	Yes	Yes	
Provider advertisement	Yes	Yes	
Distributor advertisement	Yes	Yes	
Placement opportunity	Yes	Yes	
Other: Programs, Chapters, Network, Unscheduled	Yes		Yes

Message type ID: time signal

Segmentation type ID	Base64	Cue-out, cue-in	Blackout
Provider advertisement	Yes	Yes	
Distributor advertisement	Yes	Yes	
Placement opportunity	Yes	Yes	
Other: Programs, Chapters, Network, Unscheduled	Yes		Yes

For example, read the first line in the first table as follows: When a splice insert (with no segmentation descriptor) is encountered, the base64 and cue-out, cue-in information will be inserted in the manifest; blackout information will not be inserted.

Timesignal APOS mode

Message type ID: splice insert

Segmentation type ID	Base64	Cue-out, cue-in	Blackout
No segmentation descriptor present	Yes		
Provider advertisement	Yes		
Distributor advertisement	Yes		
Placement opportunity	Yes		
Other: Programs, Chapters, Network, Unscheduled	Yes		

Message type ID: Time signal

Segmentation type ID	Base64	Cue-out, cue-in	Blackout
Provider advertisement	Yes		
Distributor advertisement	Yes		
Placement opportunity	Yes	Yes	
Other: Programs, Chapters, Network, Unscheduled	Yes		Yes

For example, read the first line in the first table as follows: When a splice insert (with no segmentation descriptor) is encountered, the base64 information will be inserted in the manifest, and cue-out, cue-in information and blackout information will not be inserted.

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 (p. 462)).

A similar feature is blackout (p. 469).

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**
> 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 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. In **Avail configuration**, set **Avail settings**, if you have not already done so:

   - SCTE-35 splice insert (default): Select this mode if the input uses splice inserts to indicate ad avails. The input might also contain messages for others events such as chapters or programs.
   - SCTE-35 time signal APOS: Select this mode if the input contains time signals of segmentation type **Placement opportunity**. The input might also contain messages for other events such as chapters or programs.

     The mode identifies which of all possible events are treated as triggers for ad avails and as triggers for blackouts. In turn, these triggers affect how manifests are decorated (p. 463), when video is blanked (p. 466), and when video is blacked out (p. 470).
3. For **Ad avail offset**, set a value, if desired. See the help for this field.
4. In **web_delivery_allowed_flag** and **no_regional_blackout_flag**, choose appropriate values. For information about these fields, see the section called "Triggers for ad avail blanking" (p. 466).

   - Follow (default): Observe the restriction and blank the content for the ad avail event.
   - Ignore: Ignore the restriction and do *not* blank the content for the ad avail event.

   Typically set both these fields to Follow. For more information about what these fields do, see the section called "Ad avail blanking restriction flags" (p. 468).
5. In **Avail blanking**, in **State**, choose **Enabled**.
6. 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" (p. 38).

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: Splice insert

Segmentation type ID	Blanked		
No segmentation descriptor present	Yes		

Segmentation type ID	Blanked		
Provider advertisement	Yes		
Distributor advertisement	Yes		
Placement opportunity	Yes		
Other: Programs, Chapters, Network, Unscheduled	No		

Message type ID: Time signal

Segmentation type ID	Blanked		
Provider advertisement	Yes		
Distributor advertisement	Yes		
Placement opportunity	Yes		
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 l ID: Splice insert

Segmentation type ID	Blanked		
No segmentation descriptor present	No		
Provider advertisement	No		
Distributor advertisement	No		
Placement opportunity	No		
Other: Programs, Chapters, Network, Unscheduled	No		

Message type ID: Time signal

Segmentation type ID	Blanked		
Provider advertisement	No		

Segmentation type ID	Blanked		
Distributor advertisement	No		
Placement opportunity	Yes		
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” (p. 466). 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 (p. 462)). For example, chapters and programs.

(A similar feature is described in the section called “Enabling ad avail blanking” (p. 465).)

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. For **Avail configuration**, set **Avail settings**, if you have not already done so:

   - SCTE-35 splice insert (default): Select this mode if the input uses splice inserts to indicate ad avails. The input might also contain messages for others events such as chapters or programs.
   - SCTE-35 time signal APOS: Select this mode if the input contains time signals of segmentation type **Placement opportunity**. The input might also contain messages for other events such as chapters or programs.

     The mode identifies which of all possible events are treated as triggers for “ad avails” and as triggers for “blackouts.” In turn, these triggers affect how manifests are decorated (p. 463), when video is blanked (p. 466), and when video is blacked out (p. 470).

3. For **Ad avail offset**, set a value, if desired. See the help for this field.
4. For **web_delivery_allowed_flag** and **no_regional_blackout_flag**, choose appropriate values. For information about these fields, see the section called “Triggers for blackout” (p. 470).

   - **Follow** (default): Observe the restriction and blank the content for the ad avail event.
   - **Ignore**: Ignore the restriction and do *not* blank the content for the ad avail event.

   Typically set both these fields to Follow. For more information about what these fields do, see the section called “Ad avail blanking restriction flags” (p. 468).
5. In **Blackout slate**, in **State**, choose **Enabled**.
6. 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” (p. 38).
7. 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.
8. For **Network end blackout**, choose **Enabled**.
9. 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” (p. 38).
10. 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 (p. 462) is not relevant. Blackout works the same with either mode.

The segmentation ID triggers blackout based on “events,” as shown in the following table.

SCTE-35 segmentation type	Blacked out
Chapter Start	Start blacking out
Chapter End	End blacking out

SCTE-35 segmentation type	Blacked out
Network Start	*End blacking out*
Network End	*Start 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 following "strength hierarchy."

SCTE-35 segmentation type	Strength
Network	1 (Strongest)
Unscheduled Event	2
Program	3
Chapter	4 (Weakest)

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.

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" (p. 466). 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 in the output

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.

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.

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 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.

Inserting SCTE-35 messages using the schedule

Use the channel schedule (p. 261) 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 *Resources: MediaLive schedule* (p. 257).

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” (p. 456) and the section called “Supported features by output type” (p. 457).

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
```

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 (p. 477).
- 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 “Step 6: Set up video” (p. 204), the section called “Step 7: Set up audio” (p. 206), and the section called “Step 8: Set up captions” (p. 208).

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 “Step 6: Set up video” (p. 204), the section called “Step 7: Set up audio” (p. 206), and the section called “Step 8: Set up captions” (p. 208).

Ingesting ancillary data from SMPTE-2038 in an MPEG-2 TS

You can configure MediaLive to extract specific ancillary data from a SMPTE-2038 PID contained in these inputs:

- HLS transport stream (TS) inputs
- MediaConnect inputs
- RTP inputs

MediaLive can extract the following data from a SMPTE-2038 that is included in one of these inputs:

- 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 – A SMPTE 12M timecode. MediaLive recognizes this timecode as an embedded timecode source.

> **Note**
> This section assumes that you are familiar with creating or editing a channel, as described in the section called “Creating a channel from scratch” (p. 144).

Well-formed SMPTE-2038 source

For MediaLive to extract and process the data appropriately, the SMPTE-2038 must meet certain criteria:

- The SMPTE-2038 must be present in every PMT.
- The PID in which the SMPTE-2038 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 stream should contain the SMPTE-2038 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.

To configure the channel to use the SMPTE-2038 data

You can specify how you want MediaLive to use SMPTE-2038 data that is in the input.

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 or if there is no SMPTE-2038, MediaLive looks for the data in other locations in the stream.
   - **Ignore** (default) – MediaLive never looks for a SMPTE-2038 PID. Even if a specific item of data is not available in other places in the stream, MediaLive doesn't look for a SMPTE-2038 PID. For example, if you set the timecode source to *embedded* and there is no timecode source in the video stream, MediaLive won't look for it in a SMPTE-2038.

How MediaLive uses the SMPTE-2038 data

If you set up to prefer SMPTE-2038 data, MediaLive handles the data as follows:

- Captions – In the **Input attachment** section for the relevant input, you might set up a captions selector that specifies **ARIB**, **Embedded**, or **Telextext**. MediaLive first looks for the specified type of captions in the SMPTE-2038. If MediaLive doesn't find the captions there, it looks in other locations in the stream.

  Regardless of where MediaLive finds the captions, when MediaLive extracts them, it processes them in the usual way, according to how you set up for captions in the output (p. 373).
- Timecode – In the **General settings** section for channel, in the **Timecode configuration** section, you might set the **Source** to **Embedded**. MediaLive first looks for a SMPTE 12M timecode in the SMPTE-2038. If MediaLive doesn't find the captions 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 on how MediaLive uses the timecode, see the section called "How timecode works at runtime " (p. 482).

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" (p. 113)
- For an HLS output – the section called "Controlling access to the output" (p. 114)

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" (p. 159)
- For frame capture outputs – the section called "Destination fields" (p. 165)
- For HLS outputs – the section called "Destination fields – Amazon S3" (p. 169)

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 and Cost Management 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
- Multiplexes
- Reservations

For information about adding and managing tags, see Managing tags (p. 480).

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 AWS Elemental 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 “Step 1: Complete channel details” (p. 146) – for information about including tags when you create a channel
- the section called “Editing and deleting a channel” (p. 213) – for information about modifying tags in an existing channel
- *Resources: MediaLive input* (p. 219) – for information about including tags in an input
- *Resources: MediaLive input security groups* (p. 243) – for information about including tags in an input security group

- the section called “Creating a multiplex and program” (p. 247) – for information about including tags in a multiplex

For information about managing tags using the AWS Elemental MediaLive API, see the following:

- Resources in the *AWS Elemental MediaLive API Reference*

Timecode configuration

You can set up the channel to include timecode metadata in the individual output encodes for any type of output group except Frame Capture. The timecode in the output is an SEI message of type `pic_timing`.

You configure timecode in two places in the channel—the input and the output.

- On the input side, you specify the source for the timecode.
- 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.

Time in the channel runs on a clock (not on a timer). The time in the output is in 24-hour format `hh:mm:ss:ff` and rolls over at midnight. For more information about the behavior of the timecode, see the section called “How timecode works at runtime ” (p. 482).

This section assumes that you are familiar with creating or editing a channel, as described in the section called “Creating a channel from scratch” (p. 144).

To configure the timecode source

1. On the **Create Channel** page, in the **General settings** section, choose **Timecode configuration**
2. For **Source**, set the appropriate value:

   - **EMBEDDED** – Use the timecode in the source video. If MediaLive doesn't find an embedded timecode in the source, it falls back to zero-based timecode (ZEROBASED).
   - **SYSTEMCLOCK**– Use the UTC time.
   - **ZEROBASED** – The time of the first frame in the output will be 00:00:00:00.

   For embedded timecode, MediaLive looks for the timecode as follows:

   - MPEG2 – A timecode inserted in each GOP header, in accordance with section 6.2.2.6 of ISO/IEC 13818-2-2000 (R2006)
   - H.264 (AVC) – 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 (HEVC) – A timecode inserted in an SEI message of type timecode, in accordance with section D.2.26 of ITU-T H.265
3. (Optional) For **Sync threshold**, enter a threshold (in frames) for synchronizing the current output timecode to the current input timecode. For information about this field, see the section called “About the synchronization threshold” (p. 482).

To include timecodes in the output

1. On the **Create Channel** page, in the **Output groups** section, create an output or choose an existing output.

2. Display the **Stream settings** section, and then choose the **Video** section.
3. For **Codec settings**, choose the codec for this video encode. More fields appear.
4. For **Timecode**, for **Timecode insertion**, choose an option:

   - **DISABLED** – This encode won't include timecode metadata.
   - **PIC_TIMING_SEI** – This encode will include timecode metadata as an SEI message of type `pic_timing`, in accordance with section D.1.2 of ISO/IEC 14496-10-2005.

Topics

About the synchronization threshold

The **Sync threshold** 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.

How timecode works at runtime

Initial channel start

When you start the channel, MediaLive samples the input timecode (if you set up the source as embedded). After that, MediaLive generates a timecode for each output frame that it produces, incrementing the timecode with each frame. The timecode isn't disrupted by an input switch (p. 402).

MediaLive looks at the input timecode again only if either of these situations occurs:

- You enabled timecode synchronization when you set up the timecode source, and a drift occurs.
- You have enabled pipeline locking and MediaLive determines a need to resynchronize the pipelines.

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.

Stopping and restarting

If you stop and restart the channel, MediaLive follows the behavior dictated for the timecode source:

- If the source is UTC clock or zero-based, MediaLive applies the relevant timecode to the first output frame.
- If the source is embedded, MediaLive samples the timecode in the input again. The situation might arise where the first time you started the channel with the source set to embedded, MediaLive didn't find an embedded timecode and therefore used zero-based. But the next time you start the channel, MediaLive might find an embedded timecode (perhaps there was an input switch). In this case, MediaLive uses that embedded timecode for the first output frame.

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 specifiction

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" (p. 132). 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 (p. 144).

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 (p. 168). Set up the video, audio, and captions outputs and encodes as you normally would (p. 204).

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" (p. 132) 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 (p. 144).

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.

Color space handling in AWS Elemental MediaLive

The source video might use a specific *color space* and a specific *brightness function*. The source video might also carry *metadata* that describes aspects of the color.

You can work with color space in two places in the channel—in the input and in the output.

Working with color space on the input side – correcting the metadata

On the input side, you can choose between two options. These options relate only to the metadata:

- You can set up to pass through any metadata to the output.
- You can set up to correct the color space metadata and pass that corrected metadata to the output.

MediaLive lets you set up each input independently.

By default, MediaLive passes through (without correction) any color space metadata that is present. It passes through both SDR and HDR metadata.

Working with color space on the output side – changing the color space and the metadata

On the output side, you can choose between three options. These options relate to the color space and its metadata:

- You can set up to pass through the content color space. MediaLive doesn't touch the color space or metadata.
- You can set up to remove the color space metadata. MediaLive doesn't touch the color space but it removes the metadata.
- You can set up to convert the color space that is in the video content to another color space, and change the color space metadata to match that conversion. MediaLive touches both the color space and the metadata.

MediaLive lets you set up each output independently.

By default, MediaLive doesn't convert the color space or change the color space metadata. It passes through the source color space and metadata to the output.

Topics

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.

HDR isn't typically associated with SD content, but MediaLive *does* support this combination.

General information about color space

Following is some general information about color space.

Definitions

The source video might use a specific *color space* and a specific *brightness function*. The source video might also carry *metadata* that describes aspects of the color.

- The color space specifies a range of pixel colors that can apply to the content.

  MediaLive supports two SDR color spaces and two HDR color spaces. For more information about the scope of the support, see the section called “How MediaLive supports the color space standards” (p. 488).
- The brightness function controls the brightness of each pixel. The brightness is also known as gamma tables, lookup tables (LUT), electro-optical transfer function (EOTF), and transfer function.
- There are three possible sets of metadata:
  - Color space metadata, which specifies which color space applies to the content. The content is said to be marked for a color space.
  - Brightness function metadata, which specifies which brightness function applies to the content.
  - Display metadata.

Color space standards

Each color space has different 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 three sets of color data for that color space.

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
601 or rec601	SDR rec. 601	BT.1886	Not applicable. This color space doesn't include display metadata.
709 or rec709	SDR 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.

Note that HDR10 and HLG use the same color space. They use different brightness functions and display metadata standards.

How MediaLive supports the color space standards

On the input side, MediaLive can read the metadata for all four color space standards. On the output side, MediaLive can produce the color space (including the metadata) for three of the color space standards.

Color space	MediaLive can read the information in the input?	MediaLive can produce the color space in the output
601	Yes	Yes
709	Yes	Yes
HDR10	Yes	Yes
HLG	Yes	No

MediaLive can't read the information in the input that comes from an AWS Elemental Link device. But when you set up the input, you can specify the color space that applies.

Configuring the handling in the input

You should identity the color spaces that are in the video content, and decide whether you want to correct the color space metadata.

Note that there can be a difference between what the color space is, and what the color space metadata states. When content is in a specific color space, the content uses a defined range of pixel colors. The color space metadata states which that range is. Ideally, the metadata matches the actual color space, so that the metadata specifies *601* and the color space actually is *601*. But in practice the metadata might be incorrect, might be missing, or might specify a color space that MediaLive doesn't support.

Topics

The procedure for input handling

> **Note**
> If you plan to remove the color space metadata from *all* the outputs, there is no point to correcting the metadata, so you can leave the default configuration for the inputs. Skip to the section called "The procedure" (p. 493).

To decide how to handle the color space metadata

1. Contact the content provider to determine what color space or color spaces apply to each video source. One individual source might contain a combination of different color spaces, particularly if the content is a VOD asset that is a few years old.

   Obtain the names of the color spaces.

   If the video source is from an AWS Elemental Link device and the color space is HDR10, also obtain the Max CLL and Max FALL values that apply to the content. (Only HDR10 has these two values.)
2. Find out from the content provider if the color space metadata is accurate. The content might be any of the following:

   - Correctly marked.
   - Incorrectly marked. This case is more common with older SDR content.
   - Unmarked (no color space metadata is present).
   - Correctly marked but unreadable. This case always applies when the input is from an AWS Elemental Link device.
   - Marked as *unknown*.
   - Marked with a color space that MediaLive doesn't support.

   One source content might be any combination of marked, unknown, and unmarked.
3. Decide if you need to add or clean up the color space metadata in each input. If you plan to include color space metadata in at least one output (either passed through from the input, or converted), then you should add it or clean it up, as appropriate.

   > **Note**
   > If you plan to remove color space metadata from all of the outputs, there is no need to clean it up. You can pass it through to the output as is. You can stop reading this procedure.

   For information on your choices for different content, see the tables in the section called "Cleanup scenarios" (p. 490).

   The following rules apply:

   - The cleanup options don't *convert* the color space metadata. They *mark it*—MediaLive inserts metadata that is missing, or it changes the metadata for incorrectly marked content.
   - Only clean up the color space if you are sure that all the unmarked portions use the color space that you choose. If the cleanup results in marking content as being in a specific color space when it isn't, then the video color quality will be degraded in the output.

- If you are planning to convert the color space on the output side, keep in mind that this conversion will apply only to marked content. So if you use cleanup to insert missing metadata, you can increase the percentage of the content that gets converted in the output.

To set up each input attached to the channel

> **Note**
> This section assumes that you are familiar with creating or editing a channel, as described in the section called "Creating a channel from scratch" (p. 144).

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**. These two fields control correction to the color space metadata.

   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	**Force**	Cleanup. MediaLive marks all the content as using the specified color space.
REC_601 or **REC_709** or HDR10 or HLG	**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.

3. This step applies only if you chose **HDR10** and the attached input is for a MediaLive device. 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.

   For **Color space settings**, choose **HDR10**. In the two fields that appear, enter the values.

Clean up scenarios for color space metadata

This section describes recommended cleanup scenarios based on the following:

- How you plan to handle the color space in the output
- The quality of the color space metadata in the input.

Scenario A – Pass through accurate metadata

The details of this scenario are the following:

- Intended handling in the output – Pass through the color space.

- Status of the input – The video content is any combination of color spaces—SDR, HDR, or both.
- Status of the input color space metadata – The metadata is correct.

Recommendation:

- **Color space** field – Set to **FOLLOW**
- **Color space usage** field – MediaLive ignores this field.

During ingest, MediaLive will retain (pass through) the metadata.

Scenario B – Convert accurate metadata

The details of this scenario are the following:

- Intended handling in the output – Convert the color space and metadata.
- Status of the input color space – The video content is any combination of color spaces—SDR, HDR, or both.
- Status of the input color space metadata – The metadata is correct.

Recommendation:

- **Color space** field – Set to **FOLLOW**
- **Color space usage** field – MediaLive ignores this field.

During ingest, MediaLive will retain (pass through) the metadata.

Scenario C – Remove metadata

The details of this scenario are the following:

- Intended handling in the output – Remove the color space metadata.
- Status of the input – The video content is any combination of color spaces—SDR, HDR, or both.
- Status of the input color space metadata – The metadata can be of any quality.

Recommendation:

- **Color space** field – Set to **FOLLOW**
- **Color space usage** field – MediaLive ignores this field.

During ingest, MediaLive will retain (pass through) the metadata. You plan to remove the metadata, so you don't care about its quality.

Scenario D – Correct the metadata

The details of this scenario are the following:

- Intended handling in the output – Convert or pass through the color space.
- Status of the input – The video content is one color space. For example, the content is all REC_601.
- Status of the input color space metadata – Some of the metadata is missing, marked as *unknown*, or marked as a color space that MediaLive doesn't support.

  In addition, some of the metadata is wrong. For example, it is marked as HDR10, but in fact, it is REC_601.

Recommendation:

- **Color space** field – Set to the color space that has unacceptable metadata.
- **Color space usage** field – Set to **FORCE**

During ingest, MediaLive will create metadata of the specified color space for all missing, unmarked, and unknown metadata.

It will also force all existing metadata to match the specified color space. Therefore, all the content in the input will be consistently marked as belonging to one color space.

Scenario E – Correct the metadata in one color space

The details of this scenario are the following:

- Intended handling in the output – Convert or pass through the color space.
- Status of the input – The video content is any combination of color spaces—REC_601, REC_709, HDR, and HLG.
- Status of the input color space metadata – The metadata for the video content of one color space is a mixture of acceptable and unacceptable. The metadata for that content is missing, marked as *unknown*, or marked as a color space that MediaLive doesn't support. But in fact, all that content should be marked as one specific color space, for example, as REC_601.

  The metadata for content for any other color space is correct. For example, the metadata for REC_709 content and HDR10 content is correct.

Recommendation:

- **Color space** field – Set to the color space that has unacceptable metadata.
- **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 will retain existing metadata.

Scenario F – Correct the metadata in multiple color spaces

The details of this scenario are the following:

- Intended handling in the output – Convert or pass through the color space.
- Status of the input – The video content is in *more than one* color space. For example, the content is a mix of REC_601, REC_709, and HDR10.
- Status of the input color space metadata – The metadata for one color space is missing, wrong, marked as *unknown*, or marked as a color space that MediaLive doesn't support. For example, the color space is REC_601, but its corresponding metadata is unreliable.

  In addition, the metadata for one or more other color spaces is also missing, wrong, unknown, or not supported. For example, the color space of that content is HLG, but its corresponding metadata is unreliable.

Recommendation:

There is no way to clean up this content because you can only mark all the content as one type of color space. But in this scenario, the metadata is incorrect in different types of color space.

If you force the color space, some of it will be forced to be correct, but some of it will be forced to incorrect information. Inaccurate metadata will result in an inaccurate conversion (if you convert in the output), or in an inferior viewing experience (if you pass through in the output).

The best recommendation we can provide is to remove the metadata on the output side—scenario C.

Configuring color space handling in each output

For each output, decide how you want to handle the color space and the color space metadata. You can do the following:

- Convert the color space in the content to a different color space in the output. MediaLive supports the following conversions:
  - Convert all content to 601 or 709. MediaLive also converts the metadata.
  - Convert all content to HDR10. This option is available only if the output codec is H.265 (HEVC).
- Remove the color space metadata. MediaLive doesn't touch the color space itself.
- Pass the color space metadata through to the output. MediaLive doesn't touch the color space itself.

> **Note**
> MediaLive converts from one color space to another based on the metadata in the content. MediaLive doesn't examine the video to try to determine whether it actually matches the color space identified in the metadata. Therefore, to successfully convert, the metadata must be as accurate as possible. To correct the metadata, see the section called “Configuring input” (p. 488).

Topics

The procedure for output handling

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.

To get ready

You must decide if you need to enable enhanced VQ mode in the output. This mode applies only to outputs that use H.264. Follow these guidelines:

Planned conversion	Details	Action
Convert to SDR	You want to convert HDR content as well as SDR content. For example, you want to convert both 601 and HLG content to 709.	You must enable enhanced VQ mode.
Convert to SDR	You *don't* want to convert any of the HDR content.	You don't need to enable enhanced VQ mode.

Planned conversion	Details	Action
	For example, you want to convert 709 to 601.	
Any action	There is no HDR10 or HLG in any of the inputs.	You don't need to enable enhanced VQ mode.
Any action	You have already enabled enhanced VQ to improve the video quality.	Leave the mode enabled.

To enable enhanced VQ mode, see the section called “Video – enhanced VQ” (p. 499).

To set up each output

Note
This section assumes that you are familiar with creating or editing a channel, as described in the section called “Creating a channel from scratch” (p. 144).

1. On the **Create channel** page, in the **Output groups** section, choose the output.
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. For **Codec settings**, choose **H264** (AVC) or **H265** (HEVC).
5. If the resolution is an SD resolution, choose **Codec details**, and then complete the **Profile**, **Tier** (for H.265 only), and **Level** fields.
6. If the resolution is an HD or UHD resolution, choose **Codec details**, and then complete **Profile**:
   - If you want to convert the content to an HDR color space, or if you want to pass through an HDR color space, choose one of the profiles that has **10BIT** in the name.
   - If you want to convert the content to an SDR color space, or if you want to pass through an SDR color space, you can choose any profile.
7. For **Color space**, choose **Color space settings**, and then choose the appropriate option to remove, pass through, or convert the color space metadata. For information about the options, see the tables in the following sections.
8. If you are converting to HDR10, you can optionally complete **Max CLL** and **Max FALL** to set display metadata. For details about a field on the MediaLive console, choose the **Info** link next to the field.

Removing color space metadata

To remove all color space metadata, in **Codec details**, expand **Color space** and set **Color space settings** to **Don't include**. The following table shows how MediaLive handles each type of color space it encounters.

Color space metadata that MediaLive encounters	How MediaLive handles the color space
Content in any color space that MediaLive supports Content with no color space metadata Content with unknown or unsupported color space metadata	MediaLive does the following for all content: • Doesn't touch the color space or brightness (the pixel values) in the output. • Removes the color space metadata.

Color space metadata that MediaLive encounters	How MediaLive handles the color space
	The output won't contain any color space metadata, brightness metadata, or display metadata.

Passing through color space

To pass through all existing color space metadata, in **Codec details**, expand **Color space** and set **Color space settings** to **Color space passthrough**. The following table shows how MediaLive handles each type of color space that it encounters.

Color space metadata 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 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).

Converting color space to SDR

To convert all the color space to 601 or 709, in **Codec details**, expand **Color space** and set **Color space settings** to **Rec601** or **Rec709**. The following table shows how MediaLive handles each type of color space it encounters.

Color space metadata 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.

Color space metadata that MediaLive encounters	How MediaLive handles the color space
	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 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. • Changes any brightness metadata to specify the new standard. • Removes any display metadata. 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 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. • Changes any brightness metadata to specify the new standard. • Removes any display metadata.

Color space metadata 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 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. • Changes any brightness metadata to specify the new standard. If you haven't enabled VQ, MediaLive doesn't convert anything. It passes through the color space metadata, and any brightness 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 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. • Changes any brightness metadata to specify the new standard.
Content marked with an unknown or unsupported color space	Doesn't touch the color space (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 (the pixel values) in the output. Leaves the content as unmarked (no color space metadata). Passes through any brightness metadata and display metadata.

Converting color space to HDR10

This option is available only when the output codec is H.265 (HEVC).

To convert all the color space to HDR10, in **Codec details**, expand **Color space** and set **Color space settings** to **HDR10**.The following table shows how MediaLive handles each type of color space it encounters.

Color space metadata 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 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. Changes the brightness metadata to specify the new standard. If you complete **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. Typically, you wouldn't choose to convert SDR to HDR if all the content in all inputs is in SDR. This is because MediaLive doesn't make any change to the pixel values of the content. A conversion from SDR to HDR would typically occur only when you have content that is both HDR and SDR and you choose to convert from HLG to HDR10. Both the HLG content and the SDR content are converted, although, as we just stated, the conversion has no effect on the color of the SDR content.
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.
Content in HLG	There is no 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.

Color space metadata that MediaLive encounters	How MediaLive handles the color space
	Change the brightness metadata to specify the new standard. If you complete **Max CLL** and **Max FALL**, MediaLive inserts the values from those fields into the display metadata.
Content marked with an unknown or unsupported color space	No change to the pixel values of the content. Leaves the content as marked with the unknown color space. Passes through any brightness metadata and display metadata.
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.

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 (p. 500).

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 HDR color space to 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 “Video – color space” (p. 486).

> **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 (p. 500).

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 the section called “Creating a channel from scratch” (p. 144).
> 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 “Step 6: Set up video” (p. 204).

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 (p. 500).
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 the section called “Creating a channel from scratch” (p. 144), and specifically with setting up the video (p. 204).

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** (p. 147) 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** (p. 147) 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** (p. 147) 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** (p. 147) 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 (p. 445) 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” (p. 56).
- Update the IAM policies for users. For more information, see the section called “Reference: summary of user access” (p. 50).
- 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” (p. 507)
   - 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” (p. 504) 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 (p. 144). It also assumes that you have read *Setup: Preparing upstream and downstream* (p. 72) 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 (p. 313).

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 (p. 445) 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 (p. 507) 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 “Supported input formats and protocols” (p. 529).
- **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 (p. 111).

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” (p. 508) 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/.

Reference: Supported sampling rate and bitrate for AAC

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 “Step 7: Set up audio” (p. 206).

> **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
	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
LC	8000	16000	28000

Profile	Sample rate (Hz)	Minimum valid bitrate (bits/sec)	Maximum valid bitrate (bits/sec)
	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
	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

Profile	Sample rate (Hz)	Minimum valid bitrate (bits/sec)	Maximum valid bitrate (bits/sec)
	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
	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

Profile	Sample rate (Hz)	Minimum valid bitrate (bits/sec)	Maximum valid bitrate (bits/sec)
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

Reference: 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

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 Package 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 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.
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

Caption	Supported in input	Supported in output	Description
			(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.
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), Subtitling Methods for Broadcast Cable.”
SMPTE-TT		Yes	Captions that are compliant with the standard "SMPTE ST 2052-1:2010"
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).

Caption	Supported in input	Supported in output	Description
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/).

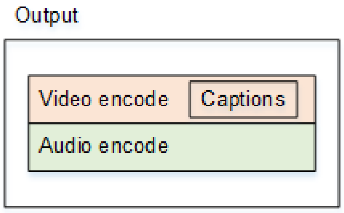
Captions categories

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
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

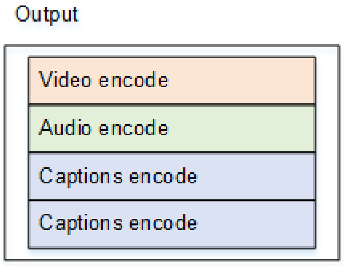
Embedded captions

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.



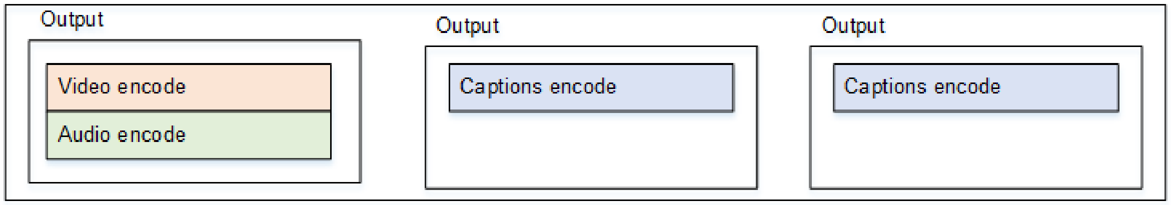
Object-style captions

All the captions encodes for a given output group are in the same output as the corresponding video and audio.



Sidecar captions

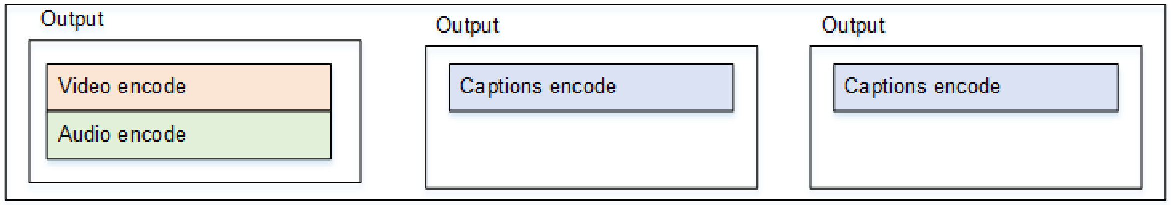
Each captions encode for a given output group is in its own "captions-only" output. The output group can contain more than one captions output, for example, one for each language.



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 (p. 519).

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” (p. 519)
- the section called “HLS or MediaPackage output” (p. 522)
- the section called “Microsoft Smooth output” (p. 523)
- the section called “RTMP output” (p. 525)
- the section called “UDP or multiplex output” (p. 526)

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

Source caption container	Source caption input	Supported output captions
		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 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

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
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
	SCTE-27	None
	Teletext	DVB-Sub Teletext

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
	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 SMPTE-TT TTML
	Teletext	Burn-in EBU-TT SMPTE-TT TTML
HLS container	Embedded	Burn-in EBU-TT-D SMPTE-TT

Source caption container	Source caption input	Supported output captions
		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
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 Embedded RTMP CaptionInfo
	SCTE-20	Embedded
Link container	Embedded	Burn-in Embedded RTMP CaptionInfo
	Teletext	None
MP4 container	Ancillary	Burn-in Embedded RTMP CaptionInfo

Source caption container	Source caption input	Supported output captions
	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
	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

Source caption container	Source caption input	Supported output captions
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 DVB-Sub Embedded Embedded+SCTE-20 SCTE-20 SCTE-20+Embedded
	Teletext	Burn-in DVB-Sub WebVTT
MP4 container	Ancillary	Burn-in DVB-Sub Embedded Embedded+SCTE-20 SCTE-20 SCTE-20+Embedded

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
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
	SCTE-27	None

Source caption container	Source caption input	Supported output captions
	Teletext	Burn-in DVB-Sub Teletext

Reference: Supported input containers and codecs

This chapter provides information about the input content types and input video and audio codecs that AWS Elemental MediaLive can ingest.

Topics

Supported input formats and protocols

The following table lists the supported input types, and describes how the input handles the source content.

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	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	Amazon S3 over a custom protocol

MediaLive input type	Push or pull?	Use case	Upstream system and supported protocol
		bucket, using a secure connection.	
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 a flow in AWS Elemental MediaConnect. This input uses a MediaConnect flow ARN, not a URI.	AWS Elemental MediaConnect over 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
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
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

MediaLive input type	Push or pull?	Use case	Upstream system and supported protocol
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

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 inputs

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
RTMP Pull	Yes	Yes
RTMP Push	Yes	No
RTP	Yes	No

Supported input class

Some inputs can be set up as either input class. Some inputs can be set up only as the single-class class. The class type to use depends on whether you plan to implement pipeline resiliency.

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

MediaLive input type	Can be set up as a single-class input	Can be set up as a standard-class input
MP4	Yes	Yes
RTMP Pull	Yes	Yes
RTMP Push	Yes	Yes
RTP	No	Yes

Support for setup as a VPC input

Some inputs can be set up in Amazon Virtual Private Cloud (Amazon VPC). For more information, see the section called “Creating an input” (p. 220).

MediaLive input type	Can be set up as a VPC input
CDI	Yes
HLS	No
Link	No
MediaConnect	No
MP4	No
RTMP Pull	No
RTMP Push	Yes
RTP	Yes

Supported codecs for inputs

The following table lists the codecs that MediaLive supports for source content.

Container	Video codecs	Audio codecs
CDI	Uncompressed video	PCM
HLS See HLS inputs (p. 534), after this table.	H.264 (AVC)	AAC Dolby Digital Dolby Digital Plus
Link	Any codec that is included in a Link container is always supported by MediaLive	Up to four stereo pairs (2.0 audio)
MediaConnect	H.264 (AVC) HEVC (H.265) MPEG-2	AAC Dolby Digital Dolby Digital Plus

Container	Video codecs	Audio codecs
		MPEG Audio PCM
MP4	H.264 (AVC) HEVC (H.265) MPEG-2	AAC
RTMP	H.264 (AVC)	AAC
RTP	H.264 (AVC) HEVC (H.265) MPEG-2	AAC Dolby Digital Dolby Digital Plus MPEG Audio PCM

HLS inputs

The audio and video assets can be multiplexed into a single stream or in a separate audio rendition group. If you are using audio in a rendition group, it can be selected by using the **Group ID** and **Name** that is in the **#EXT-X-MEDIA** tag.

Characteristics of video and audio sources

Orientation

MediaLive only ingests landscape video. If a video source is configured as portrait, MediaLive will ingest it but will rotate it to landscape.

Other characteristics

Container	Video characteristics	Audio characteristics
CDI	• Uncompressed 4:2:2 8-bit • Uncompressed 4:2:2:2 10-bit	24-bit Big-Endian PCM Mono (1.0), Dual mono (2.0), Stereo (2.0), 5.1, 7.1 222, SGRP 48kHz, 96 kHz

Reference: Supported output containers and codecs

This chapter provides information about the output content types and output video and codecs that MediaLive can produce.

Topics

Supported containers and downstream systems

The following table lists the output formats and protocols that MediaLive supports.

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 (p. 536), after this table.	Amazon S3, over a custom protocol	No	Yes. A channel can contain only one archive output group.
Frame Capture	Send a series of JPEG files to an Amazon S3 bucket. See Amazon S3 Bucket Names (p. 536), 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 container or an fMP4 container	Send an HLS stream to a server that supports HTTP PUT or WebDav.	HTTP 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 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	Yes	No
HLS with a standard container	Send an HLS stream to a	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
or an fMP4 container	container on MediaStore.			
HLS with a standard container or an fMP4 container	Send an HLS stream to an Amazon S3 bucket. See Amazon S3 Bucket Names (p. 536), 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
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 to VPC

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" (p. 503).

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
Frame Capture	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 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 codecs for outputs

The following table lists the codecs that MediaLive supports.

Container	Video codecs	Audio codecs
Archive	H.264 (AVC)	AAC Dolby Digital

Container	Video codecs	Audio codecs
		Dolby Digital Plus MPEG-1 Layer II
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 Dolby Digital Plus
HLS with an fMP4 container	H.265 (HEVC)	AAC Dolby Digital Dolby Digital Plus
MediaPackage	H.264 (AVC) H.265 (HEVC)	AAC Dolby Digital Dolby Digital Plus
Microsoft Smooth	H.264 (AVC) H.265 (HEVC)	AAC Dolby Digital Dolby Digital Plus
Multiplex	H.264 (AVC) H.265 (HEVC)	AAC Dolby Digital Dolby Digital Plus
RTMP or RTMPS	H.264 (AVC)	AAC Dolby Digital Dolby Digital Plus
UDP	H.264 (AVC) H.265 (HEVC)	AAC Dolby Digital Dolby Digital Plus

Reference: 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.
`$w$`	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 identifier. The modifier pads the resolved value with leading zeros. The format is`%0n`, where *n* is a number.

Identifier	Format	Description
		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

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. This content includes 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 user accounts with 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 recommend TLS 1.2 or later.
- 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 personal 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 fields such as a **Name** field. This includes when you work with AWS Elemental MediaLive or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form 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.

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” (p. 213)
- *Resources: MediaLive devices* (p. 215)
- the section called “Deleting an input” (p. 242)
- the section called “Deleting an input security group” (p. 245)
- the section called “ Deleting multiplexes, programs, and channels ” (p. 249)
- the section called “Deleting a reservation” (p. 256)

Identity and access management in 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.

For more information about identity and access management for MediaLive, see *Setting up: IAM permissions* (p. 14) and *Setting up: IAM permissions for production* (p. 21).

Compliance validation for AWS Elemental MediaLive

Third-party auditors assess the security and compliance of AWS Elemental MediaLive as part of multiple AWS compliance programs. These include ISO and HIPAA.

For a list of AWS services in scope of specific compliance programs, see AWS Services in Scope by Compliance Program. 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 MediaLive is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides resources to help with compliance:

- Security and Compliance Quick Start Guides – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- Architecting for HIPAA Security and Compliance Whitepaper – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- AWS Config – This AWS 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 that helps you check your compliance with security industry standards and best practices.

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 the AWS global network security procedures that are described in the Amazon Web Services: Overview of Security Processes whitepaper.

You use AWS published API calls to access MediaLive through the network. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). 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**

update-history-change	update-history-description	update-history-date
Audio in an HLS input (p. 533)	We have updated HLS inputs to clarify that MediaLive now supports ingest of audio in rendition groups.	September 10, 2021
Akamai digest authentication (p. 186)	The user guide now documents correct settings when using digest authentication for an Akamai destination.	September 1, 2021
Maintenance windows (p. 336)	The user guide now documents the recently added maintenance windows feature. This allows a day and time to be set for required, routine maintenance to channels.	August 26, 2021
Support for font styles with WebVTT captions (p. 363)	The user guide now documents the recently implemented feature to support passing the color and position style from the source captions to the output, when the output captions are WebVTT.	August 11, 2021
Extracting SCTE-35 message information from manifest (p. 459)	MediaLive now supports the ability to extract ad avail information from the manifest of an HLS input, and convert the information into SCTE-35 messages of type splice insert. You can then set up the channel to include those messages in outputs.	August 5, 2021
WebVTT output captions (p. 514)	MediaLive now supports the ability to produce WebVTT output captions by converting object-based source captions that are that are in DVB-Sub or SCTE-27 format. The section on supported captions and the section on working with captions have been revised.	June 18, 2021
Color space (p. 486)	We have revised the guidance for handling color space. We have	June 15, 2021

	also added information about ingesting color space metadata from an AWS Elemental Link device.	
Addition to constraints: AWS Elemental Link UHD input charges (p. 9)	The user guide includes information about the charges for ingesting inputs from the type of device.	June 15, 2021
Amazon CloudWatch metrics (p. 316)	MediaLive now supports CloudWatch metrics. These metrics are published in the MediaLive namespace in CloudWatch.	June 1, 2021
Pipeline locking with UDP outputs (p. 444)	MediaLive now supports pipeline locking with UDP outputs.	May 26, 2021
Source video orientation (p. 534)	The user guide now clarifies that MediaLive only supports landscape video orientation.	May 25, 2021
Addition to constraints: input types for input switching (p. 9)	The user guide includes clarifications about the types of VOD inputs you can include in input switching.	May 25, 2021
Trick-play in MediaPackage output groups (p. 484)	You can now include a trick-play track in a MediaPackage output group.	May 4, 2021
Sharing and cloning output encodes (p. 476)	There is a new section in the guide to describe share or clone encodes within and among outputs in the channel.	May 4, 2021
Planning encode sharing (p. 139)	The section about designing encode outputs has been revised to provide guidance for sharing or cloning encodes within and among outputs in the channel.	May 4, 2021
Pipeline locking (p. 442)	There is a new section in the guide to describe how the existing pipeline locking feature works.	May 4, 2021
Revision to resiliency (p. 452)	We have revised the information on resiliency options in the channel. Several sections of the guide have been revised.	April 13, 2021
Clarification about properties that control output charges (p. 204)	The section on setting up the video encode in the channel now includes information about the fields (resolution, framerate, bitrate) that contribute to the calculation of charges for each output.	April 13, 2021

Calculation of output charges with QVBR rate control (p. 500)	There is now information about how to set the bitrate in order to ensure accurate output charges when using QVBR rate control.	April 13, 2021
Support for font styles with TTML captions (p. 363)	The user guide now documents the recently implemented feature to support passing the font styles from the source captions to the output, when the output captions are TTML.	April 12, 2021
Motion graphics overlay (p. 432)	AWS Elemental MediaLive now supports the ability to insert one motion graphics overlay on the video in a channel.	April 5, 2021
Support for Amazon S3 access control lists (ACLs) (p. 479)	The guide now includes information about the recently added support for ACLs when delivering outputs to Amazon S3.	March 30, 2021
Access required for the workflow wizard (p. 50)	We have added information about the access that users of the workflow wizard require. To use the workflow wizard, users need access to additional AWS services.	March 15, 2021
MediaLive workflow wizard (p. 309)	The console now includes a workflow wizard that lets you quickly create a functioning channel.	March 3, 2021
Support for automatic input failover with CDI inputs (p. 384)	MediaLive now supports setting up two CDI inputs as automatic input failover pairs. To implement automatic input failover with CDI inputs, you must set up the two CDI inputs as *partner* inputs.	February 15, 2021
Addition to constraints: maximum number of audio selectors (p. 9)	The user guide now specifies the maximum number of audio input selectors in one channel.	February 3, 2021
Maximum frame capture outputs and encodes (p. 9)	Expansion of limits on the maximum number of frame capture encodes.	February 2, 2021
Image Media Playlist specification for trick-play (p. 484)	You can now include a trick-play track in an HLS output group. This track follows the Image Media Playlist specification, version 0.4.	February 2, 2021

Input clipping (p. 394)	There is now support for clipping a file input that is stored on an HTTP server.	February 1, 2021
Captions in CDI inputs (p. 84)	CDI inputs can now include captions.	February 1, 2021
Reorganization (p. 529)	The section on containers and codecs is now in two sections, one for input containers and codecs, and one for output containers and codecs.	January 27, 2021
New feature—Delivery via your VPC (p. 503)	You can now set up a channel to have output endpoints in Amazon Virtual Private Cloud (Amazon VPC).	January 27, 2021
Quotas section moved (p. 8)	We have moved the chapter about quotas to the start of the guide.	December 11, 2020
Pricing section moved (p. 7)	We have moved the information about pricing out of the What Is chapter into its own chapter at the start of the guide.	December 11, 2020
Permissions for device transfers (p. 27)	The information about requirements for user access has been updated to include access to the operations for transferring devices. In this section, look for the term *Devices*.	December 4, 2020
MediaLive devices (p. 349)	There is no longer a rule that each channel can have only one Elemental Link input attached to it. This means that you can now include Elemental Link inputs in a standard channel (a channel with two pipelines).	December 4, 2020
Device transfer (p. 216)	The user guide has been updated to describe how to transfer a device to another AWS account.	December 4, 2020
Revision to documentation for creating output groups (p. 158)	As part of the revision to the section about planning the workflow with the downstream system, we have revised the sections about configuring the destination in each type of output group.	November 9, 2020

Revision to documentation for creating inputs (p. 219)	As part of the revision to the section about planning the workflow with the upstream system, we have revised the procedures for creating each type of input.	November 9, 2020
Revision of information for designing the workflow (p. 72)	The section about designing the workflow has been revised and reorganized.	November 9, 2020
Revision of information for designing the channel (p. 121)	The section about designing the channel has been revised and reorganized.	November 9, 2020
AAC audio codec (p. 510)	We have added a section that lists the supported combinations of coding mode, profile, sampling rate, and bitrate for the AAC codec.	November 5, 2020
Addition to constraints: maximum number of schedule actions (p. 9)	The user guide now specifies the maximum number of actions that the channel schedule can contain.	October 2, 2020
Access tables (p. 50)	We have moved two tables to the top level of the chapter that describes how to set up IAM for production. The tables are now easier to find when you are browsing.	September 29, 2020
Audio-only outputs (p. 339)	There is a new section in the guide to describe how to set up an output group that contains only audio.	September 4, 2020
EBU-TT-D output captions (p. 514)	MediaLive now supports EBU-TT-D captions in outputs. The section on supported captions and the section on working with captions have been revised.	July 24, 2020
Support for input prepare (p. 259)	The channel schedule now supports the ability 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.	June 18, 2020
Full support for immediate mode for schedule actions (p. 257)	The channel schedule now immediate start type for all actions in the schedule.	June 18, 2020
Quotas for Elemental Link inputs (p. 8)	The table for MediaLive quotas now includes the quotas for Elemental Link inputs.	May 29, 2020

Permissions for devices (p. 27)	The information about requirements for user access has been updated to include access to devices. In this section, look for the term *Devices*.	May 29, 2020
MediaLive devices (p. 349)	AWS Elemental MediaLive now supports the ability to ingest content that is streaming from an AWS Elemental Link hardware device.	May 29, 2020
Ancillary data in SMPTE-2038 (p. 477)	AWS Elemental MediaLive now supports the ability to extract specific ancillary data from a SMPTE-2038 PID in specific inputs.	May 29, 2020
AWS Elemental Devices (p. 215)	AWS Elemental MediaLive now includes the MediaLive device, as a representation of an AWS Elemental Link hardware device.	May 29, 2020
Enhanced VQ mode (p. 499)	AWS Elemental MediaLive now supports the ability to set up enhanced VQ mode in any H.264 video encode	May 6, 2020
Color space (p. 486)	AWS Elemental MediaLive now supports converting HDR color space to SDR color space in the outputs.	May 6, 2020
SCTE-35 schedule actions in follow mode (p. 257)	The channel schedule now supports the ability to enter actions to insert a SCTE-35 message in follow mode. You set up the action to follow an existing input switch action. MediaLive inserts the action when the input for that input switch ends.	April 24, 2020
Revision to documentation for creating output groups (p. 158)	The procedures for creating an output group and an output have been revised and reorganized.	April 23, 2020
Redundant HLS manifests (p. 426)	There is a new section in the guide to describe the recently added support for redundant HLS manifests.	April 23, 2020
Custom manifests in HLS output groups (p. 422)	There is a new section in the guide to describe the existing feature of customizing the paths for HLS manifests.	April 23, 2020

Section for changing channel class (p. 214)	The information on changing the channel class has been moved, to make it easier to find. The information is now in its own subsection in the MediaLive Channel section.	April 3, 2020
Resiliency section (p. 452)	The MediaLive Features section now includes a subsection that lists the various features that let you improve resiliency in the MediaLive channel. Several sections of the guide have been revised.	April 3, 2020
New feature—Automatic input failure (p. 351)	There is now support for setting up two inputs as an input failover pair, to provide resiliency in the channel when there are problems upstream of MediaLive.	April 3, 2020
Channel class (p. 386)	The MediaLive Features section now includes a subsection about channel class, which controls pipeline redundancy. Previously, the information about this existing feature was harder to find in the guide.	April 3, 2020
Timecode configuration (p. 481)	The user guide now includes information on configuring timecode, including the supported standards for timecodes in the input and for timecodes in the output.	March 26, 2020
Audio rendition groups for HLS outputs (p. 341)	There is a new section in the guide to describe the existing feature of audio rendition groups in HLS outputs.	March 3, 2020
Input switching limitations (p. 406)	The rule that you can't have both MediaConnect inputs and VPC inputs attached to one channel no longer applies. That rule has been removed from the list of limits.	February 2, 2020
Number of frame capture outputs (p. 537)	A channel can now contain up to three frame capture output groups. Previously, it could contain only one.	January 31, 2020

New variable data identifier: $sn $ inserts the segment number of the video in the output. (p. 538)	AWS Elemental MediaLive supports a variable data identifier `$sn$` to insert the ability to insert ID3 tags in every segment in HLS and MediaPackage outputs.	December 31, 2019
ID3 segment tagging (p. 390)	AWS Elemental MediaLive now supports the ability to insert ID3 tags in every segment in HLS and MediaPackage outputs.	December 31, 2019
Summary of rules and constraints (p. 9)	The user guide now includes a section that describes some of the rules and constraints that apply to AWS Elemental MediaLive.	December 12, 2019
IAM access requirements for multiplexes (p. 27)	The list of operations required for the multiplex feature has been revised to include ec2.DescribeAvailabilityZones.	December 12, 2019
Nielsen watermarks (p. 441)	AWS Elemental MediaLive now supports the abilty to convert Nielsen watermarks to ID3 metadata.	December 9, 2019
Revision to documentation for schedules (p. 537)	The channel schedule now supports the ability to enter actions to switch the channel from ingesting one input to another input. This chapter has been revised.	December 8, 2019
SCTE-35 schedule actions in immediate mode (p. 257)	The channel schedule now supports the ability to enter actions to insert a SCTE-35 message immediately. It previously supported insertion only at a fixed time.	November 25, 2019
SCTE-35 schedule actions in immediate mode (p. 537)	The channel schedule now supports the ability to enter actions to insert a SCTE-35 message immediately. It previously supported insertion only at a fixed time.	November 25, 2019
Permissions for MediaLive multiplex (p. 27)	The information about requirements for user access has been updated to include access to the multiplex feature.	November 25, 2019
Multiplex and MPTS (p. 434)	You can now create a multiplex to produce an MPTS.	November 25, 2019

Color space (p. 486)	AWS Elemental MediaLive now includes options for converting color space and for handling video color space metadata.	September 12, 2019
Ad avails: manifest decoration in HLS (p. 458)	Now, in an HLS output group, you can enable manifest decoration but disable SCTE-35 passthrough. (Previously, you had to disable both features or enable both features in the same output group.)	September 12, 2019
Support for an immediate input switch (p. 257)	The channel schedule now supports the ability to enter a request to switch an input as soon as possible, which is immediate mode.	July 25, 2019
Support for an immediate input switch (p. 258)	The channel schedule now supports the ability to enter a request to switch an input as soon as possible (immediate mode).	July 25, 2019
Input clipping (p. 394)	You can now clip a file input so that MediaLive ingests only a portion of the file. The clipping instructions are included in the input switch action that you set up in the schedule.	July 25, 2019
Dynamic inputs (p. 387)	The input switching feature of the MediaLive schedule now supports switching to a dynamic input. With a dynamic input, you specify a different file each time you switch to the input.	July 25, 2019
Security information (p. 541)	The new Security chapter contains information about how to configure AWS Elemental MediaLive to meet your security and compliance objectives.	May 28, 2019
Reorganization of the user guide (p. 1)	The chapters in the user guide have been reorganized and renamed. No sections have been removed. The "components" of MediaLive have been sorted so that they appear together (channel, input, and so on), and some sections have been moved into the chapter called "MediaLive Features".	May 28, 2019

Quotas for VPC inputs (p. 8)	The table for MediaLive quotas now includes the quotas for VPC inputs.	May 3, 2019
Changing the channel class (p. 213)	You can now change the channel class for an existing channel	May 3, 2019
Changing the channel class (p. 214)	You can now change the channel class for an existing channel.	May 3, 2019
Standard channels and single-pipeline channels (p. 146)	You can now set up a channel with a single-pipeline. The **Create channel** page includes a **Channel class** field.	April 5, 2019
Revision to documentation for setting up access (p. 21)	The procedures for setting up the user and the service with AWS IAM access have been revised and split into two chapters, Setting Up> and Setting Up for Production.	March 22, 2019
Permissions for trusted entity for VPC delivery (p. 56)	The information on permissions for the trusted entity has been updated to include the actions required to support the VPC delivery feature.	March 22, 2019
MediaPackage output group (p. 192)	There is a new output group type: MediaPackage, which lets your send output to AWS Elemental MediaPackage. This new output group type creates an HLS output that is streamlined for delivery to MediaPackage.	March 13, 2019
Support for pausing and unpausing a channel pipeline (p. 262)	The channel schedule now supports the ability to enter actions to pause and unpause a pipeline in a channel.	March 8, 2019
Support for encrypted HLS source (p. 79)	MediaLive now supports ingest of encrypted HLS sources.	March 5, 2019
Push inputs from your VPC (p. 219)	You can create an RTP push input or an RTMP push input to push content from an upstream system that is in your Amazon VPC to MediaLive. Several chapters have been updated, including the chapter about inputs and the chapter about creating a channel from scratch.	February 20, 2019
Quotas for reservations (p. 8)	The table for MediaLive quotas now includes the quotas for reservations.	February 11, 2019

Tagging (p. 479)	There is now support for attaching tags to channels, inputs, input security groups, multiplexes, and reservations.	February 8, 2019
Frame capture output group (p. 164)	There is a new output group type: frame capture output group, to send a series of frame capture files to Amazon Simple Storage Service.	January 25, 2019
Integration with AWS CloudTrail (p. 333)	There is now support for logging MediaLive API calls with CloudTrail.	January 18, 2019
Integration with AWS Elemental MediaConnect (p. 219)	You can set up to use a flow from AWS Elemental MediaConnect as an input for a channel. Information has been added to the chapter about inputs and the chapter about creating a channel from scratch.	December 7, 2018
Revision to documentation for schedules (p. 257)	The channel schedule now supports the ability to enter actions to switch the channel from ingesting one input to another input. This chapter has been revised.	November 8, 2018
Input switching (p. 402)	There is now support for input switching. The channel must be set up with multiple inputs, and the schedule must contain actions to switch from one input to another. A new chapter on input switching has been added. In addition, information about multiple inputs (which are required for input switching) has been added in the existing chapters on planning and creating a channel.	November 8, 2018
Change in input quotas (p. 8)	Quotas for inputs are now split into three categories: push and pull.	November 8, 2018
QVBR rate control mode feature (p. 500)	There is a new option for the rate control field that is part of setting up the video encode in the outputs of a channel. The option is "quality-defined variable bitrate" (QVBR).	October 17, 2018
Doc-only update, Introduced RSS feed (p. 544)	You can now subscribe to RSS feeds for notifications of updates to this user guide.	June 21, 2018

Change	Description	Date
Requirement for redundant channels in AWS Elemental MediaPackage	If your downstream system is AWS Elemental MediaPackage, there is no longer a requirement to set up two channels in that service. The sections the section called "Step 2: Set up the downstream system" (p. 64) and the section called "Step 7: Coordinate with downstream systems" (p. 111) have been revised to reflect this change.	August 31, 2018
Schedule feature	There is now support for adding SCTE-35 messages and static image overlays to the channel's schedule, for inclusion in a running channel. The sections *Resources: MediaLive schedule* (p. 257) and the section called "Image overlays" (p. 392) have been added.	August 24, 2018
Reservations feature	There is now support for purchasing a reservation for processing. With a reservation, you pay a lower rate on specific processing. The section *Resources: MediaLive reservations* (p. 251) has been added.	June 19, 2018
Channel logs feature	There is now support for sending log information to Amazon CloudWatch Logs. The sections the section called "Logging" (p. 158) and the section called "Monitoring using CloudWatch Logs" (p. 330) have been added. The section *Setting up: IAM permissions for production* (p. 21) has been revised to include setup for logs.	June 13, 2018
Console alerts feature	There is now support for viewing channel alerts on the console. See *Monitoring a channel or multiplex* (p. 313). The information on setting up for alerts has been moved to this chapter.	June 6, 2018
Support for RTMP outputs	the section called "Step 7: Coordinate with downstream systems" (p. 111) now	April 18, 2018

Change	Description	Date
	includes RTMP. The section the section called “RTMP output group” (p. 199) has been added. the section called “Captions categories” (p. 517) now includes RTMP CaptionInfo. the section called “RTMP output” (p. 525) has been added.	
Input edit feature, Input security group edit feature	Editing of an existing input and an editing of an existing input security group are now supported. See the section called “Editing an input” (p. 241) the section called “Editing an input security group” (p. 244)	March 23,2018
Input delete feature, Input security group delete feature	Deleting an input and deleting an input security group are now supported. See the section called “Editing an input” (p. 241) the section called “Editing an input security group” (p. 244)	March 23, 2018
Creating a channel	the section called “Creating a channel from scratch” (p. 144) has been extensively revised, particularly with more information and examples on setting up destinations for output groups.	March 23, 2018
Input specification feature	Input specification fields ensure that the service allocates sufficient processing resources and correctly calculates processing charges. See the section called “Step 2: Attach inputs” (p. 148).	February 15, 2018
Channel edit feature	Editing of the fields in an existing (saved) channel is now supported. See the section called “Editing a channel” (p. 213).	February 15, 2018
Custom template feature	Users can create custom templates from existing channels, and can import those templates into new channels. See the section called “Creating a channel from a template or by cloning” (p. 210).	February 15, 2018
New service and guide	This is the initial release of *AWS Elemental MediaLive User Guide*.	November 27, 2017

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 AWS Elemental 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 AWS Elemental 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.
- AWS Elemental 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 General Reference*.