# Table of Contents

 What Is AWS Elemental MediaLive? ................................................................. 1
 How MediaLive Works ................................................................................. 1
   Pipelines ................................................................................................. 2
   MediaLive Inputs ..................................................................................... 2
   MediaLive Channels ............................................................................... 2
   MediaLive Schedule .............................................................................. 3
 Terminology ............................................................................................. 4
 Related Services ....................................................................................... 5
 Accessing AWS Elemental MediaLive ...................................................... 6
 Pricing ........................................................................................................ 6
 Feature Rules and Limits ........................................................................... 8
 Setting Up: IAM Permissions ..................................................................... 11
 Signing Up ................................................................................................. 11
 Creating an Administrator IAM User ....................................................... 11
 Creating a Non-Administrator IAM User .................................................. 13
   Step 1: Create Customer Managed Policies ........................................... 14
   Step 2: Create an IAM Group ................................................................. 15
   Step 3: Create or Add an IAM User to Your Group ................................. 16
 Setting up as a Trusted Service ................................................................. 17
 Setting Up: IAM Permissions for Production .......................................... 18
 Setting Up Administrators and Users ....................................................... 19
 Creating an Administrator User with Limited Access ............................ 19
 Creating a Non-Administrator User .......................................................... 22
   Step 1: Requirements for Permissions .................................................. 23
   Summary of Step 1 Access Requirements .............................................. 33
   Step 2: Identify Categories of Users ....................................................... 35
   Step 3: Create the Custom Policies ....................................................... 36
   Step 4: Create the Groups .................................................................... 36
   Step 5: Create Users ............................................................................. 37
   Step 6: Setting Up Required Data ......................................................... 39
 Setting Up AWS Elemental MediaLive as a Trusted Service .................. 39
   How the Trusted Entity Is Created and Attached ................................. 40
   Creating Trusted Entity Roles ............................................................... 40
   Setting Up Permissions for Non-Administrator Users ........................... 46
 Getting Started Tutorial .......................................................................... 49
   Prerequisites ......................................................................................... 49
   Step 1: Set Up the Upstream System .................................................... 49
   Step 2: Set Up the Downstream System ............................................... 50
   Step 3: Create an Input ....................................................................... 50
   Step 4: Set up Key Information ............................................................ 51
   Step 5: Attach the Input ..................................................................... 51
   Step 6: Set up Input Video, Audio, Captions ...................................... 51
   Step 7: Create an HLS Output Group .................................................. 52
   Step 8: Set Up the Output and Encodes .............................................. 53
   Step 9: Create Your Channel ............................................................... 53
   Step 10: Start the Upstream System and the Channel ......................... 53
   Step 11: Clean Up ................................................................................ 54
 Components of AWS Elemental MediaLive ........................................... 55
   Inputs .................................................................................................... 55
   Channels .............................................................................................. 55
   Encodes .............................................................................................. 55
   Outputs ............................................................................................... 56
   Output Groups ...................................................................................... 56
 Input Security Groups .............................................................................. 56
# AWS Elemental MediaLive User Guide

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>How Components Are Associated</td>
<td>56</td>
</tr>
<tr>
<td>Setting up: Planning Your workflow</td>
<td>57</td>
</tr>
<tr>
<td>Assessing the video source</td>
<td>57</td>
</tr>
<tr>
<td>Identifying Resiliency Requirements</td>
<td>58</td>
</tr>
<tr>
<td>Setting up the upstream system</td>
<td>58</td>
</tr>
<tr>
<td>MediaConnect push</td>
<td>59</td>
</tr>
<tr>
<td>MP4 pull</td>
<td>59</td>
</tr>
<tr>
<td>RTP pull</td>
<td>59</td>
</tr>
<tr>
<td>RTMP push</td>
<td>60</td>
</tr>
<tr>
<td>RTMP pull</td>
<td>60</td>
</tr>
<tr>
<td>HLS pull</td>
<td>61</td>
</tr>
<tr>
<td>Planning the inputs</td>
<td>61</td>
</tr>
<tr>
<td>Planning the channel</td>
<td>62</td>
</tr>
<tr>
<td>Rules for encodes in an output</td>
<td>63</td>
</tr>
<tr>
<td>Examples of channel designs</td>
<td>63</td>
</tr>
<tr>
<td>Non-ABR asset with captions embedded in the video</td>
<td>63</td>
</tr>
<tr>
<td>Non-ABR segmented asset with captions as separate objects</td>
<td>64</td>
</tr>
<tr>
<td>Non-ABR segmented asset with captions as sidecars</td>
<td>64</td>
</tr>
<tr>
<td>ABR asset with captions embedded in the video</td>
<td>64</td>
</tr>
<tr>
<td>ABR asset with captions in sidecars</td>
<td>65</td>
</tr>
<tr>
<td>Setting up the downstream system</td>
<td>65</td>
</tr>
<tr>
<td>For an Archive output</td>
<td>65</td>
</tr>
<tr>
<td>For an HLS output</td>
<td>66</td>
</tr>
<tr>
<td>For a MediaPackage output</td>
<td>67</td>
</tr>
<tr>
<td>For a Frame Capture output</td>
<td>67</td>
</tr>
<tr>
<td>For a Microsoft Smooth output</td>
<td>67</td>
</tr>
<tr>
<td>For an RTMP or RTMPS output</td>
<td>68</td>
</tr>
<tr>
<td>For a UDP output</td>
<td>68</td>
</tr>
<tr>
<td>Next steps</td>
<td>68</td>
</tr>
<tr>
<td>MediaLive Channels</td>
<td>69</td>
</tr>
<tr>
<td>Creating a channel from scratch</td>
<td>69</td>
</tr>
<tr>
<td>Getting ready</td>
<td>70</td>
</tr>
<tr>
<td>Step 1: Complete channel details</td>
<td>71</td>
</tr>
<tr>
<td>Step 2: Attach inputs</td>
<td>73</td>
</tr>
<tr>
<td>Step 3: Complete input settings</td>
<td>77</td>
</tr>
<tr>
<td>Step 4: Complete general settings</td>
<td>79</td>
</tr>
<tr>
<td>Step 5: Create output groups</td>
<td>80</td>
</tr>
<tr>
<td>Step 6: Set up video</td>
<td>116</td>
</tr>
<tr>
<td>Step 7: Set up audio</td>
<td>117</td>
</tr>
<tr>
<td>Step 8: Set up captions</td>
<td>118</td>
</tr>
<tr>
<td>Step 9: Save Channel</td>
<td>118</td>
</tr>
<tr>
<td>Creating a Channel from a Template or by Cloning</td>
<td>118</td>
</tr>
<tr>
<td>About Templates</td>
<td>119</td>
</tr>
<tr>
<td>About Cloning</td>
<td>120</td>
</tr>
<tr>
<td>Creating a Channel from a Template</td>
<td>120</td>
</tr>
<tr>
<td>Creating a Channel by Cloning</td>
<td>120</td>
</tr>
<tr>
<td>Creating a Custom Template</td>
<td>121</td>
</tr>
<tr>
<td>Editing and Deleting a Channel</td>
<td>121</td>
</tr>
<tr>
<td>Editing a Channel</td>
<td>121</td>
</tr>
<tr>
<td>Editing the Tags Associated with a Channel</td>
<td>122</td>
</tr>
<tr>
<td>Deleting a Channel</td>
<td>122</td>
</tr>
<tr>
<td>Updating Channel Class</td>
<td>122</td>
</tr>
<tr>
<td>Changing the channel to a single-pipeline channel</td>
<td>122</td>
</tr>
<tr>
<td>Changing the channel class to standard</td>
<td>123</td>
</tr>
<tr>
<td>Viewing a Channel Configuration</td>
<td>124</td>
</tr>
<tr>
<td>MediaLive Input</td>
<td>126</td>
</tr>
<tr>
<td>Categories for inputs</td>
<td>126</td>
</tr>
</tbody>
</table>
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 (p. 1)
- AWS Elemental MediaLive Terminology (p. 4)
- Related Services (p. 5)
- Accessing AWS Elemental MediaLive (p. 6)
- Pricing (p. 6)

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 Setting up: Planning Your workflow (p. 57).

**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 MediaLive Schedule (p. 159)

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.

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. 73).
- 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. 116). 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/.
# 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. 348).

<table>
<thead>
<tr>
<th>Resource or feature</th>
<th>Constraint or rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input media</td>
<td>The input must include video. MediaLive can’t ingest source content that doesn’t include video. The source content can contain video, or it can contain video and audio. In addition, it can contain captions, but captions are always optional. For information about input types, see the section called “Supported Input Types and Upstream Systems” (p. 336).</td>
</tr>
<tr>
<td>Input number, live inputs</td>
<td>You can attach 0 to 2 live inputs to a channel, either as push inputs, pull inputs, or both. Note that if you set up a failover pair (to implement automatic input failover (p. 218)), you use up this quota and can’t attach another live input to the channel.</td>
</tr>
<tr>
<td>Input number, file inputs</td>
<td>You can attach up to 20 inputs to a channel. After you have counted the live inputs, the remainder can be file inputs. File inputs are always pull inputs.</td>
</tr>
</tbody>
</table>
| Input types – in multiple-input channels | These restrictions apply to the different input types that can be in the "pool" of inputs attached to one channel:  
• You can’t have both MediaConnect inputs and VPC inputs attached to one channel.  
• 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, including both RTP VPC inputs and RTMP VPC push inputs. But all those inputs must be in the same two Availability Zones. |
<p>| Output, types | Maximum of one archive output groups in a channel. For information about output types, see the section called “Supported Containers and Downstream Systems” (p. 339). |</p>
<table>
<thead>
<tr>
<th>Resource or feature</th>
<th>Constraint or rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum of three frame capture output groups in a channel.</td>
<td>For information about output types, see the section called “Supported Containers and Downstream Systems” (p. 339). A channel that includes a UHD output can include only one UHD output, 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. 348). The maximum number of outputs in this type of channel is a limitation. You can't change it.</td>
</tr>
<tr>
<td>Standard definition (SD) video is supported with all codecs. For information about supported output codecs, see the section called “Supported Codecs for Outputs” (p. 341). High definition (HD) video is supported with H.264 and H.265. Ultra-high definition (UHD or 4K) video is supported with H.265.</td>
<td>For information about output video resolutions, see the section called “Supported Codecs for Outputs” (p. 341).</td>
</tr>
<tr>
<td>Each multiplex produces only one MPTS.</td>
<td>For information about multiplex, see the section called “Multiplex and MPTS” (p. 290). All multiplex programs must include video.</td>
</tr>
<tr>
<td>Maximum of 20 programs per multiplex.</td>
<td>Each program in a multiplex is single use. It is attached only to one multiplex, and you can use it only for that multiplex.</td>
</tr>
<tr>
<td>Each channel contains one and only one output group, of type multiplex. It can't contain any other type of output group.</td>
<td>Each channel is single use. You can attach it to only one program in the multiplex. You can use it only for that multiplex.</td>
</tr>
<tr>
<td>Resource or feature</td>
<td>Constraint or rule</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Image Overlays</td>
<td>Maximum of eight different overlays (layers) active at one time in a channel. This means that the video can show up to eight different overlays at the same time. For information about image overlay, see the section called “Image Overlays” (p. 260).</td>
</tr>
<tr>
<td>Schedule and input switches</td>
<td>The schedule can contain any number of scheduled input switching actions. For information about input switching, see the section called “Input Switching” (p. 262). You can switch to a specific input as many times as you want.</td>
</tr>
<tr>
<td>Frequency of API requests</td>
<td>5 steady-state TPS (transactions per second)</td>
</tr>
<tr>
<td></td>
<td>30 burst TPS</td>
</tr>
</tbody>
</table>
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 Tutorial (p. 49).

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. 18).

Topics
- Signing Up for AWS Elemental MediaLive (p. 11)
- Creating an Administrator IAM User (p. 11)
- Creating a Non-Administrator IAM User (p. 13)
- Setting up AWS Elemental MediaLive as a Trusted Service (p. 17)

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

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:
Creating an Administrator IAM User

• 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. 17).

**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. 18).

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.

![Diagram showing the relationship between a policy, a group, and a user.]

**Topics**

- Step 1: Create Customer Managed Policies (p. 14)
- Step 2: Create an IAM Group (p. 15)
- Step 3: Create or Add an IAM User to Your Group (p. 16)
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
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
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
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. 14).

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)
   - CloudWatchReadOnlyAccess (AWS managed policy)
   - AmazonEC2FullAccess (AWS managed policy for access to AWS Virtual Private Network)
   - AWS Elemental MediaPackageFullAccess (AWS managed policy)
   - ResourceGroupsandTagEditorFullAccess (AWS managed policy)
   - AmazonSSMFullAccess (AWS managed policy for access to AWS Systems Manager)
   - AmazonSQSFullAccess (AWS managed policy)
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. 17) 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. 15), 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. 15).

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. 15).
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. 14).
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. 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. 11) 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. 13) 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 (p. 19)
• Creating an Administrator User with Limited Access (p. 19)
• Creating a Non-Administrator User (p. 22)
• Setting Up AWS Elemental MediaLive as a Trusted Service (p. 39)

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. 11).

• 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. 19).

• 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. 22).

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. 23).
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.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Corresponding Service in IAM</th>
<th>Type of Access</th>
<th>Actions to Include in the Policy</th>
</tr>
</thead>
</table>
| 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. 11), you created this policy in the the section called “Step 1: Create
<table>
<thead>
<tr>
<th>Feature</th>
<th>Corresponding Service in IAM</th>
<th>Type of Access</th>
<th>Actions to Include in the Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitoring Channel Health</td>
<td>CloudWatch</td>
<td>Limited access to CloudWatch (the same access as non-administrator users).</td>
<td>See the section called “Summary of Step 1 Access Requirements” (p. 33).</td>
</tr>
<tr>
<td>Setting Up for Email</td>
<td>CloudWatch Events</td>
<td>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.)</td>
<td>Use the managed policy CloudWatchEventsFullAccess. The administrator might not need all these actions, but giving full access is probably low risk.</td>
</tr>
<tr>
<td>Setting Up Channel Logging</td>
<td>CloudWatch Logs</td>
<td>Limited access to CloudWatch Logs (the same access as non-administrator users).</td>
<td>See the section called “Summary of Step 1 Access Requirements” (p. 33).</td>
</tr>
<tr>
<td>Creating a VPC Input</td>
<td>EC2</td>
<td>Limited access to Amazon EC2 (the same access as non-administrator users)</td>
<td>See the section called “Summary of Step 1 Access Requirements” (p. 33).</td>
</tr>
<tr>
<td>Setting Up User Identities</td>
<td>IAM</td>
<td>Limited access to manage users, groups, policies, and trusted entity roles.</td>
<td>The action ChangePassword And all actions that have any of these strings in their name: &quot;User&quot;, &quot;Group&quot;, &quot;Policy&quot;, &quot;Policies&quot;, &quot;Role&quot;, &quot;AccessKey&quot;, &quot;LoginProfile&quot;. Except don’t include actions that also have the string &quot;Instance&quot;, or the string &quot;ContextKeys&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Setting Up Email</td>
<td>SNS</td>
<td>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.)</td>
<td>Use the managed policy AmazonSNSFullAccess. The administrator might not need all these actions, but giving full access is probably low risk.</td>
</tr>
</tbody>
</table>
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. 23). Then, based on your decisions, modify or create policies and groups in the usual way.

Summary of Steps

To createIAM 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.
Step 1: Requirements for Permissions

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.

Step 1: Identify Requirements for Permissions for Users

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:
Step 1: Requirements for Permissions

- Permissions to create – Permissions to create, modify, and delete channels, inputs, input security groups, or reservations
- Permissions to view – Permissions to view the details of channels, inputs, input security groups, and reservations
- Permissions to run – Permissions to start and stop channels

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 *MediaLive Reservations* (p. 153).

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

<table>
<thead>
<tr>
<th>Permissions</th>
<th>Service Name in IAM</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create, Modify, and Delete Channels, Inputs, and Input Security Groups</td>
<td>MediaLive</td>
<td>One or more write operations</td>
</tr>
<tr>
<td>View Channels, Inputs, and Input Security Groups</td>
<td>MediaLive</td>
<td>One or more list operations</td>
</tr>
<tr>
<td>Create or Modify Multiplexes</td>
<td>MediaLive</td>
<td>CreateMultiplex</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DescribeMultiplex</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ListMultiplexes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UpdateMultiplex</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EC2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DescribeAvailabilityZones</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You need this operation to view the list of Availability Zones on the MediaLive console, so that you can choose two for the multiplex.</td>
</tr>
<tr>
<td>Delete Multiplexes</td>
<td>MediaLive</td>
<td>DeleteMultiplex</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DescribeMultiplex</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ListMultiplexes</td>
</tr>
<tr>
<td>View Multiplexes</td>
<td>MediaLive</td>
<td>DescribeMultiplex</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ListMultiplexes</td>
</tr>
<tr>
<td>Change the Class for a Channel</td>
<td>MediaLive</td>
<td>UpdateChannelClass</td>
</tr>
<tr>
<td>Run Channels</td>
<td>MediaLive</td>
<td>StartChannel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StopChannel</td>
</tr>
<tr>
<td>Run Multiplexes</td>
<td>MediaLive</td>
<td>StartMultiplex</td>
</tr>
</tbody>
</table>
### Permissions

<table>
<thead>
<tr>
<th>Permissions</th>
<th>Service Name in IAM</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attach Tags to Channels, Inputs, and Input Security Groups When Creating Those Resources</td>
<td>MediaLive</td>
<td>CreateTag</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DeleteTags</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ListTagsForResources</td>
</tr>
<tr>
<td>Create, Modify, Delete, and View Reservations and Offerings</td>
<td>MediaLive</td>
<td>DeleteReservation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DescribeOffering</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DescribeReservation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ListOfferings</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ListReservations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PurchaseOffering</td>
</tr>
</tbody>
</table>

### Requirements for AWS CloudTrail

AWS Elemental 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.

<table>
<thead>
<tr>
<th>Permissions</th>
<th>Service Name in IAM</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Metrics</td>
<td>CloudWatch</td>
<td>ListMetrics</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetMetricData</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetMetricStatistics</td>
</tr>
</tbody>
</table>

### 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 a Channel or Multiplex Using CloudWatch” (p. 202).)

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.

<table>
<thead>
<tr>
<th>Permissions</th>
<th>Service Name in IAM</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write</td>
<td>CloudWatch Events</td>
<td>All actions</td>
</tr>
<tr>
<td>Write</td>
<td>SNS</td>
<td>All actions</td>
</tr>
</tbody>
</table>

**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 a Channel Using CloudWatch Logs” (p. 205).

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.

<table>
<thead>
<tr>
<th>Permissions</th>
<th>Service Name in IAM</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Logs</td>
<td>CloudWatch Logs</td>
<td>FilterLogEvents</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GetLogEvents</td>
</tr>
<tr>
<td>Set Retention Policy</td>
<td>CloudWatch Logs</td>
<td>DeleteRetentionPolicy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PutRetentionPolicy</td>
</tr>
</tbody>
</table>

**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. 127).
Step 1: Requirements for Permissions

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

<table>
<thead>
<tr>
<th>Permissions</th>
<th>Service Name in IAM</th>
<th>Actions</th>
</tr>
</thead>
</table>
| View the VPC subnets and VPC security groups on the MediaLive console | EC2 | DescribeSubnets
| | | DescribeSecurityGroups

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. 13) 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 (p. 27)
- Options for Implementing the Role (p. 28)
- Requirements for Permissions for the Simple Option (p. 29)

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."
Step 1: Requirements for Permissions

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. 29).

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. 39).

Requirements for Permissions for the Simple Option

Read this section if you decide that the simple option (p. 28) 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. 39).)

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.

You must set up users as follows:
Step 1: Requirements for Permissions

- 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.

<table>
<thead>
<tr>
<th>Permissions</th>
<th>Service Name in IAM</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Choose the <strong>Create role from template option</strong></td>
<td>IAM</td>
<td>CreateRole, PutRolePolicy, AttachRolePolicy</td>
</tr>
<tr>
<td>Choose <strong>MediaLiveAccessRole</strong> from the list in <strong>Use existing role</strong></td>
<td>IAM</td>
<td>ListRole, PassRole</td>
</tr>
<tr>
<td>Choose <strong>Update</strong></td>
<td>IAM</td>
<td>GetRolePolicy, PutRolePolicy, AttachRolePolicy</td>
</tr>
</tbody>
</table>

Requirements for AWS Elemental MediaConnect

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

Users don't need special permissions to specify an MediaConnect flow as input.

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. 66). (Note that both MediaLive and MediaPackage have "channels"; however, they are different objects.)

Regardless of the type of output group, users don't need special permissions to specify a MediaPackage channel as the destination for output.

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
Users don't need special permissions to specify a MediaStore container as the destination for output.

**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. 23).
- 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.

<table>
<thead>
<tr>
<th>Permissions</th>
<th>Service Name in IAM</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select</td>
<td>Systems Manager</td>
<td>GetParameters</td>
</tr>
<tr>
<td>Create</td>
<td>Systems Manager</td>
<td>PutParameter</td>
</tr>
<tr>
<td>Modify and Delete</td>
<td>Systems Manager</td>
<td>DeleteParameter, DeleteParameters, DescribeParameters, GetParameter, GetParameterHistory, GetParameters, GetParametersByPath</td>
</tr>
</tbody>
</table>

Summary of Step 1 Access Requirements

The following table shows all the types of permissions that you might need to assign to users.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Corresponding Service in IAM</th>
<th>Tasks</th>
<th>Actions to Include in the Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>MediaLive Features</td>
<td>MediaLive</td>
<td>Create, modify, and delete channels, inputs, and input security groups</td>
<td>One or more List operations</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>One or more Read operations</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>One or more Write operations</td>
</tr>
<tr>
<td>MediaLive</td>
<td>View channels, inputs, and input security groups</td>
<td>One or more List operations</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>One or more Read operations</td>
</tr>
<tr>
<td></td>
<td>Run channels</td>
<td>StartChannel</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>StopChannel</td>
<td></td>
</tr>
</tbody>
</table>
## Summary of Step 1 Access Requirements

<table>
<thead>
<tr>
<th>Feature</th>
<th>Corresponding Service in IAM</th>
<th>Tasks</th>
<th>Actions to Include in the Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>MediaLive</td>
<td>Create, modify, delete, and view offerings and reservations</td>
<td>DeleteReservation, DescribeOffering, DescribeReservation, ListOfferings, ListReservations, PurchaseOffering</td>
<td></td>
</tr>
<tr>
<td>MediaLive</td>
<td>Attach tags when creating a resource</td>
<td>CreateTags, DeleteTags, ListTagsForResource</td>
<td></td>
</tr>
<tr>
<td>Monitoring Channel Health</td>
<td>CloudWatch</td>
<td>ListMetrics, GetMetricData, GetMetricStatistics</td>
<td></td>
</tr>
<tr>
<td>Setting Up Events</td>
<td>CloudWatch Events</td>
<td>All actions</td>
<td>The managed policy CloudWatchEventsFullAccess provides these permissions</td>
</tr>
<tr>
<td>Setting Up Channel Logging</td>
<td>Amazon CloudWatch Logs</td>
<td>View logs</td>
<td>FilterLogEvents, GetLogEvents</td>
</tr>
<tr>
<td>Creating a VPC input</td>
<td>EC2</td>
<td>View the VPC subnets and VPC security groups on the MediaLive console</td>
<td>DescribeSubnets, DescribeSecurityGroups</td>
</tr>
<tr>
<td>Simple Option for the Trusted Entity Role</td>
<td>IAM</td>
<td>Create the MediaLiveAccessRole</td>
<td>CreateRole, PutRolePolicy, AttachRolePolicy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Choose the MediaLiveAccessRole</td>
<td>ListRole, PassRole</td>
</tr>
</tbody>
</table>
### 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. 23), 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:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Corresponding Service in IAM</th>
<th>Tasks</th>
<th>Actions to Include in the Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Systems Manager</td>
<td>Systems Manager</td>
<td>Create a password parameter using the MediaLive console or the AWS Systems Manager console</td>
<td>DeleteParameter</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DeleteParameters</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DescribeParameters</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GetParameter</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GetParameterHistory</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GetParameters</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GetParametersByPath</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PutParameter</td>
</tr>
<tr>
<td>Setting Up Email Notification</td>
<td>Amazon SNS</td>
<td>All actions</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The managed policy AmazonSNSFullAccess provides these permissions</td>
<td></td>
</tr>
</tbody>
</table>

### AWS Systems Manager

Choose a password parameter from the dropdown list on the MediaLive console

DescribeParameters
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. 23). 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. 35). See http://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.

![Group with Policies](image)

**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 (p. 37)
- Modify an Existing User (p. 38)

**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. 36)) 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. 36)).

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. 319). 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. 36).

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. 11) 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. 36). 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/](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 a Channel or Multiplex Using CloudWatch” (p. 202) for instructions.
- For the MediaLive role, if you chose the simple option (p. 28), make sure users know that the only role they will ever choose is the **MediaLiveAccessRole**.

If you set up for the complex option (p. 28), 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. 28) 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. 27) and the section called “Options for Implementing the Role” (p. 28).

Topics
• How the Trusted Entity Is Created and Attached (p. 40)
• Creating Trusted Entity Roles (p. 40)
• Setting Up Permissions for Non-Administrator Users (p. 46)

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. 28).

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. 40).
• 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. 46).

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. 28).

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. 27).

Topics
• Step 1: Determine the Access Requirements (p. 41)
• Step 2: Create Policies (p. 44)
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 at the bottom of this section 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.

**Summary Of Requirements for the MediaLive Trusted Entity**

The following table lists the services that are often used in MediaLive deployment. The third column specifies whether MediaLive itself needs access to these services. If yes, then the fourth column suggests an existing policy that provides the required access, or else lists the operations and resources that you would typically include in a custom policy.
<table>
<thead>
<tr>
<th>Service</th>
<th>Tasks</th>
<th>Type of Access Required</th>
<th>Suggested Actions or Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Elemental MediaLive</td>
<td>Working with MediaLive features.</td>
<td>MediaLive doesn't need access to itself. Only the users need access.</td>
<td></td>
</tr>
<tr>
<td>AWS CloudTrail</td>
<td>Capturing MediaLive activity.</td>
<td>MediaLive doesn't need IAM access for this task.</td>
<td></td>
</tr>
<tr>
<td>CloudWatch</td>
<td>Displaying CloudWatch metrics information on the console, to monitor channel health.</td>
<td>MediaLive doesn't need IAM access for this task. Only the users need access.</td>
<td></td>
</tr>
<tr>
<td>CloudWatch Events and Amazon SNS</td>
<td>Setting up email notification so that users can be notified about MediaLive alerts that are sent to CloudWatch Events.</td>
<td>MediaLive doesn't need access for this task. Only the users need access.</td>
<td></td>
</tr>
<tr>
<td>CloudWatch Logs</td>
<td>Sending channel log information to CloudWatch Logs when a channel is running.</td>
<td>When the channel is running, MediaLive must be able to send log messages to CloudWatch Logs.</td>
<td>CreateLogGroup, CreateLogStream, PutLogEvents, PutMetricFilter, PutRetentionPolicy, DescribeLogStreams, DescribeLogGroups, And these resources: arn:aws:logs:* arn:aws:log-group:*</td>
</tr>
<tr>
<td>Amazon EC2</td>
<td>Creating an RTP VPC input or RTMP VPC push input.</td>
<td>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.</td>
<td>CreateNetworkInterface, CreateNetworkInterfacePermissions, DescribeNetworkInterfaces, DescribeSecurityGroup, DescribeSubnets</td>
</tr>
<tr>
<td></td>
<td>Deleting an RTP VPC input or RTMP VPC push input.</td>
<td>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.</td>
<td>DeleteNetworkInterface, DeleteNetworkInterfacePermissions, DescribeNetworkInterfaces, DescribeSubnets</td>
</tr>
<tr>
<td>Service</td>
<td>Tasks</td>
<td>Type of Access Required</td>
<td>Suggested Actions or Policy</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MediaConnect</td>
<td>Creating a MediaConnect input.</td>
<td>When the user creates an MediaConnect input, MediaLive must have read/write access to the MediaConnect flow, in order to add an output to that flow.</td>
<td>ManagedDescribeFlow ManagedAddOutput To include these actions that start with &quot;Managed&quot; 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.</td>
</tr>
<tr>
<td></td>
<td>Deleting a MediaConnect input.</td>
<td>When the user creates an 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.</td>
<td>ManagedDescribeFlow ManagedDeleteOutput To include these actions that start with &quot;Managed&quot; 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.</td>
</tr>
<tr>
<td></td>
<td>Creating a MediaConnect entitlement.</td>
<td>MediaLive doesn't need access for this task.</td>
<td></td>
</tr>
<tr>
<td>AWS Elemental</td>
<td>Sending channel output to MediaPackage when a channel is running, if your deployment uses this service.</td>
<td>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.</td>
<td>DescribeChannel</td>
</tr>
<tr>
<td>MediaPackage</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Creating Trusted Entity Roles

<table>
<thead>
<tr>
<th>Service</th>
<th>Tasks</th>
<th>Type of Access Required</th>
<th>Suggested Actions or Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Elemental MediaStore</td>
<td>Sending and retrieving assets from a MediaStore container when a channel is running, if your deployment uses this service.</td>
<td>When the channel is running, MediaLive must have read access (for a source) or read/write access (for a destination).</td>
<td>ListContainers, DescribeObject, PutObject, GetObject, DeleteObject</td>
</tr>
<tr>
<td>Resource Group Tagging</td>
<td>Attaching tags when creating resources—channels, inputs, and input security groups —and revising tags on existing resources.</td>
<td>MediaLive doesn't need IAM access for this task. Only the users need access.</td>
<td></td>
</tr>
<tr>
<td>Amazon S3</td>
<td>Sending and retrieving assets from an Amazon S3 bucket when a channel is running, if your deployment uses this service.</td>
<td>When the channel is running, MediaLive must have read access (for a source) or read/write access (for a destination) to the buckets.</td>
<td>ListBucket, PutObject, GetObject, DeleteObject</td>
</tr>
<tr>
<td>AWS Systems Manager</td>
<td>Creating a password parameter on the MediaLive console.</td>
<td>MediaLive doesn't need IAM access for this task. Only the users need access.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Using a password parameter in the channel configuration. See the section called “AWS Systems Manager Parameter Store” (p. 31).</td>
<td>When the channel is running, MediaLive must have read access to the AWS Systems Manager Parameter Store.</td>
<td>The managed policy AmazonSSMReadOnlyAccess</td>
</tr>
</tbody>
</table>

### Step 2: Create Policies

Any person who is an administrator can create a policy.

In the section called “Step 1: Determine Requirements” (p. 41), 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*.

**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. 41), 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/](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. 319). 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. 28). 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**.

The policy document should now look like this:

```json
{
  "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. 39).

**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. 28).

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. 40).
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.

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.

<table>
<thead>
<tr>
<th>Permissions</th>
<th>Service Name in IAM</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attach</td>
<td>IAM</td>
<td>iam:PassRole</td>
</tr>
</tbody>
</table>

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. 28).

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. 36) 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. 47).
2. In the group that you created or will create for regular users (see the section called “Step 4: Create the Groups” (p. 36)), include this policy.
3. Identify any other policies that relate to the IAM service, and detach them from this group.
Getting Started with AWS Elemental MediaLive

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 master 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.

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. 11), 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://39fu04.mediapackage.us-east-1.amazonaws.com/in/v1/88dpe/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. 49) 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, 
\texttt{rtp://198.51.100.10:5000} and \texttt{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. 53).

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.

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. 50). For example, https://39fu4.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. 31).

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. 50). 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 in 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 in the section called “Step 6: Set up Input Video, Audio, Captions” (p. 51).
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. 49). Set them up to push to the two destinations that you noted in the section called “Step 3: Create an Input” (p. 50). 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. 145) 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 “Reference: Supported Containers and Codecs” (p. 336).

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 MediaLive Input Security Groups (p. 145).

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.
Setting up: Planning your workflow for AWS Elemental MediaLive

To use AWS Elemental MediaLive to transcode a video asset, you follow this basic workflow:

1. Assess the video source to ensure that MediaLive can accept it.
2. Identify your resiliency requirements.
3. Set up the upstream system that will provide the video source to MediaLive.
4. Create an input for your video asset.
5. Optionally associate the input with an input security group (required only for certain types of inputs).
6. Create a channel in which you identify the input to transcode and specify how MediaLive should ingest and encode that input.
7. Start (run) your channel. MediaLive ingests the input, encodes the input (and any associated audio, captions, and metadata), and then creates the output.
8. Send the output to a downstream system—for example, send the output to an origin service or a packager such as AWS Elemental MediaPackage.

Topics
- Assessing the video source (p. 57)
- Identifying Resiliency Requirements (p. 58)
- Setting up the upstream system (p. 58)
- Planning the channel (p. 62)
- Examples of channel designs (p. 63)
- Setting up the downstream system (p. 65)
- Next steps (p. 68)

Assessing the video source

The upstream system is the system that streams the video to AWS Elemental 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 stadium at a sports event. The upstream system can be on the public internet or in a VPC that you created in Amazon Virtual Private Cloud (Amazon VPC).

Assess the video source to ensure that it delivers what you expect:

- Make sure that the upstream system is capable of streaming to your MediaLive input using one of the supported protocols. See the section called “Supported Input Types and Upstream Systems” (p. 336).
- Determine if your chosen protocol uses a “push” or a “pull.” With a push protocol, the upstream system pushes to MediaLive. With a pull protocol, MediaLive pulls from the upstream system. See the section called “Supported Input Types and Upstream Systems” (p. 336).
- Find out how the input—the video codecs, audio codecs, and captions formats—is encoded:
  - Make sure that MediaLive supports the input video codecs and the input audio codecs. See the section called “Supported Input Types and Upstream Systems” (p. 336).
• Make sure that MediaLive supports the input captions formats. Additionally, make sure that MediaLive supports conversion of that input captions format to the output captions format that you want in your specific output types. See the section called “Reference: Supported Captions” (p. 325).

Identifying Resiliency Requirements

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.

Automatic input failover

You can set up two 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.

For more information about automatic input failover, see the section called “Automatic input failover” (p. 218).

Pipeline redundancy

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

You set up for pipeline redundancy by setting up the channel as a standard channel. The channel has two pipelines—pipelines 0 and 1. Both pipelines ingest, process the source content, and produce output. The downstream system handles output from one of the pipelines and ignores the other. 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.

If you don't want pipeline redundancy, you set up the channel as a single-pipeline channel. If that single pipeline fails, output to the downstream system stops.

For more information on pipeline redundancy, see the section called “Channel class” (p. 243).

Combining automatic input failover and pipeline redundancy

You can combine both automatic input failover and pipeline redundancy. For more information, see the section called “About automatic input failover” (p. 219).

Setting up the upstream system

You must set up your upstream system before you start working in AWS Elemental MediaLive.

Topics

• MediaConnect push (p. 59)
• MP4 pull (p. 59)
• RTP push (p. 59)
MediaConnect push

MediaLive can accept a flow from AWS Elemental MediaConnect as an input:

- The AWS Elemental MediaConnect flow and the MediaLive input must be in the same AWS Region. If possible, set up AWS Elemental MediaConnect and MediaLive in the same Region. If that is not possible, then set up a distribution in AWS Elemental MediaConnect to move the source to the same Region as the MediaLive input.
- If you are setting up two or more flows in MediaConnect (to support resiliency (p. 58)), make sure that the streams in the two flows are identical in terms of resolution and bitrate.
- Keep in mind that MediaConnect must be pushing its flow or flows to the MediaLive input before you start the channel. All push inputs are live inputs, and a live input must be already pushing even if it is not the first input in the channel.

For detailed information about setting up MediaConnect and MediaLive, see the section called “Creating a MediaConnect push input” (p. 129).

MP4 pull

- If you are setting up two sources (to support resiliency (p. 58)), the upstream system must provide two video streams. For optimized redundancy, MediaLive runs each source on different encoder pipelines in different Availability Zones. You don't have to set up these Availability Zones because MediaLive does it for you.
- If you are setting up two sources, make sure that the two files are identical in terms of resolution and bitrate.
- Keep in mind that the video source must be ready to be pulled before the channel starts to ingest the input.
  - If this input is the only input or the first input in the channel, it must be ready before you start the channel.
  - If this input is not the first input, it must be ready approximately 30 seconds before the channel switches to this input.

RTP push

- Make sure that the source is a streaming source, not a file source. A push input works only with a streaming source.
- We recommend that you enable FEC in the source. A source that includes FEC is less likely to result in an output that has visual disruptions.
- Make sure that the upstream system is set up to send over RTP, not UDP. The UDP protocol is not supported as an input into MediaLive.
- If you are setting up two sources (to support resiliency (p. 58)), the upstream system must provide two video streams. For optimized redundancy, MediaLive runs each source on different encoder pipelines in different Availability Zones. You don't have to set up these Availability Zones because MediaLive does it for you.
• If you are setting up two sources, make sure that the two streams are identical in terms of resolution and bitrate.
• Keep in mind that the upstream system must be pushing the video source to the input before you start the channel. All push inputs are live inputs, and a live input must be already pushing even if it is not the first input in the channel.
• This information applies if the upstream system is on the public internet. From the upstream system, obtain the IP addresses that the two streams will push from and make a note of them. You will need this information to set up the required input security groups for the MediaLive inputs that you will create.
• This information applies if the upstream system is in a VPC that you created in Amazon VPC, and you plan to follow the typical plan of creating a VPC input in MediaLive. From the upstream system, obtain the IP addresses of the two streams that the upstream system will push, and make a note of them. You will need this information when you identify the VPC security groups for the MediaLive inputs that you will create.

RTMP push

• Make sure that the source is a streaming source, not a file source. A push input works only with a streaming source.
• If you are setting up two sources (to support resiliency (p. 58)), the upstream system must provide two video streams. For optimized redundancy, MediaLive runs each source on different encoder pipelines in different Availability Zones. You don't have to set up these Availability Zones because MediaLive does it for you.
• If you are setting up two sources, make sure that the two streams are identical in terms of resolution and bitrate.
• Determine the application name and application instance for this video source.

The upstream system might have already assigned these names. These names might be provided to you separately (for example, the application name is livestream and the application instance is curling) or as a path (livestream/curling). Make a note of these names.

If the upstream system has not assigned names, you could request names that work for you. We recommend that you use live as the application name and a name of your choosing as the application instance. Make sure that you and the operator of the upstream system agree on these names.
• Keep in mind that the upstream system must be pushing the video source to the input before you start the channel. All push inputs are live inputs, and a live input must be already pushing even if it is not the first input in the channel.
• From the upstream system, obtain the IP addresses that the two streams will push from and make a note of them. You will need this information to set up the required input security groups for the MediaLive inputs that you will create.

RTMP pull

• If you are setting up two sources (to support resiliency (p. 58)), the upstream system must provide two video streams. For optimized redundancy, MediaLive runs each source on different encoder pipelines in different Availability Zones. You don't have to set up these Availability Zones because MediaLive does it for you.
• If you are setting up two sources, make sure that the two streams are identical in terms of resolution and bitrate.
• For a pull input, keep in mind that the video source must be ready to be pulled before you start the channel. This rule applies for both HLS VOD inputs and HLS live inputs.
HLS pull

- If you are setting up two sources (to support resiliency (p. 58)), the upstream system must provide two video streams. For optimized redundancy, MediaLive runs each source on different encoder pipelines in different Availability Zones. You don’t have to set up these Availability Zones because MediaLive does it for you.
- If you are setting up two sources, make sure that the two streams are identical in terms of resolution and bitrate.
- The sources can be encrypted or unencrypted. For information on setting up encrypted content, see later in this section.
- For a pull input, keep in mind that the video source must be ready to be pulled before you start the channel. This rule applies for both HLS VOD inputs and HLS live inputs.

Handling encrypted source content

MediaLive can ingest an HLS input that is encrypted according to the HTTP Live Streaming specification. MediaLive supports AES-128 but not AES-SAMPLE. MediaLive supports encryption using either static or rotating keys.

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 HLS manifests on the upstream system (an HTTP(S) server, an AWS Elemental MediaStore container, or an Amazon S3 bucket), and places the encryption key on the license server. When the channel that contains this input starts, MediaLive obtains the main manifest, reads the #EXT-X-KEY tag for the URL of the encryption key, and obtains the encryption key from that location. MediaLive decrypts the input using the encryption key and the IV.

Get ready

Contact the upstream system and verify that:

- The content is encrypted with AES-128.
- The manifest includes the #EXT-X-KEY tag with these attributes:
  - The METHOD attribute specifies AES-128
  - The URL specifies the license server for the encryption key.
  - The IV is blank or specifies the IV to use. If the IV is blank, MediaLive uses the value in 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.

After you ensure that the upstream system is set up correctly, there is no further setup for you to perform. There is no special setup to perform in the input or in the channel. When the channel starts, MediaLive reads the manifest and determines from the #EXT-X-KEY that the input is encrypted. It follows the process described earlier in this section to decrypt the content as it is ingested.

Planning the inputs

To plan your input or inputs, identify which individual video, audio, and captions assets you want to extract from each input and which ones that you want to omit. For example, you must extract one video file, but you can choose to omit some captions languages.
The rules for extracting input are the following:

- You must extract one and only one video file from each input.

If the channel will have multiple inputs, there is no requirement for the video properties in the various inputs to be identical in terms of codec, resolution, frame rate, color space, scan type, and so on. So, for example, the video in one input might be HEVC, while the video in another input might be H.264.

- You can extract zero or more audio files from each input. Typically, you extract multiple audio files so that you can include multiple languages in the output. But you can also extract multiple audio files to extract different audio formats, for example, AAC and Dolby Digital.

If the channel will have multiple inputs, there is no requirement for the audio files in the various inputs to be identical in terms of codec, sample rate, bitrate, and so on.

- You can extract zero or more captions files.

If the channel will have multiple inputs, read the information about setting up for captions in the section called “Input Switching” (p. 262).

Planning the channel

To plan your channel, follow these guidelines:

1. Identify the output protocols (for streaming outputs) or the number of different output file types (for archive and frame capture outputs).

   For example, you could create a streaming ABR HLS output asset, a streaming non-ABR HLS output asset, and an archive version of the HLS output asset (containing the highest bitrate video). You could also create a streaming ABR Smooth output asset.

2. For the first output asset, identify the number of video encodes that you need:

   - Some output assets consist of one video encode (one set of encoding settings). In this case, you should plan to create an output group that contains one video output.
   - An ABR output asset will have more than one video encode, for example, one high-bitrate video, one medium-bitrate video, and one low-bitrate video. The encoding instructions are identical (for example, they all use H.264) except for the bitrate. In this case, you should plan to create an output group with more than one video output.

3. For the first output asset, identify the audio encodes that you need.

   Typically, you need one encode for each language (English, French, and so on).

4. For the first output asset, identify the captions that you need.

   Typically, you need one encode for each captions language (English, French, and so on).

5. Group these encodes into outputs, as described in the section called “Examples of channel designs” (p. 63). Make sure that the groupings follow the rules for encodes in outputs (p. 63).

6. Group the outputs into one output group. For example, group the outputs for the ABR HLS outputs into one HLS output group.

7. Repeat the design of encodes, outputs, and output group for each output asset.

Here’s the result of running a channel that follows this design:

- Each output that you create in the channel becomes one media asset.
- If a media asset includes manifests, one master manifest is created for each output group and one “variant manifest” is created for each output.
Rules for encodes in an output

The following rules apply to the organization of encodes into outputs:

- **Video** – Each video encode goes in its own output. One output can't contain two videos.
- **Audio** – In a non-ABR asset, each audio encode goes in its own output. In an ABR asset, the audio encodes don't go in their own output; they are contained in the same output as the video encode.
- **Captions** – Embedded-type captions encodes always go in the same output as the video encode, inside (embedded) in the video encode. Object-style captions encodes go in their own object in the same output as the video encode. Sidecar captions encodes always go in their own output.

These rules mean that one output can contain the following:

- A video encode
- A video encode and one or more audio encodes
- A video encode and one embedded-type captions
- A video encode and one or more object-type captions
- A video encode, one or more audio encodes, and one embedded-type captions
- A video encode, one or more audio encodes, and one or more object-type captions
- An audio encode
- A sidecar-type captions

Examples of channel designs

The following sections show examples of channel designs. The designs progress from simple designs that contain only one video offering to more complex designs for adaptive bitrate streaming (ABR) assets that contain several video offerings.

**Topics**

- **Non-ABR asset with captions embedded in the video** (p. 63)
- **Non-ABR segmented asset with captions as separate objects** (p. 64)
- **Non-ABR segmented asset with captions as sidecars** (p. 64)
- **ABR asset with captions embedded in the video** (p. 64)
- **ABR asset with captions in sidecars** (p. 65)

**Non-ABR asset with captions embedded in the video**

For a non-ABR asset, you create the following outputs and encodes in the channel: one output that contains one video asset, as many audio assets as you require, and as many captions assets as you require.

The following illustration shows one output in one output group. The output contains one video asset, one captions asset, and two audio assets.
Running this channel produces one segmented media file that contains the video, captions, and audio encodes.

In outputs that have manifests, it also produces one manifest file and one variant manifest file.

Non-ABR segmented asset with captions as separate objects

In this example, captions are separate objects, but they are contained within the media asset (they are not sidecar files). Captions such as DVB-Sub are set up as separate objects.

Running this channel produces one segmented media file that contains the video, audio, and captions encodes.

Non-ABR segmented asset with captions as sidecars

The following illustration shows a non-ABR asset with one video asset and two audio assets in one output group, and two captions assets, each in their own output.

Running this channel produces one media file that contains video and audio, and one media file for each captions asset.

ABR asset with captions embedded in the video

For an ABR asset, you create the following outputs and encodes:

- Several video outputs, each containing one video encode (for example, one high-bitrate video, one medium-bitrate video, and one low-bitrate video) and the same embedded captions encode.
- One or more audio encodes (for example, one for each language).
The following illustration shows an example configuration.

```
<table>
<thead>
<tr>
<th>Output Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output V</td>
</tr>
<tr>
<td>Output V</td>
</tr>
<tr>
<td>Output V</td>
</tr>
<tr>
<td>Output A</td>
</tr>
<tr>
<td>Output A</td>
</tr>
</tbody>
</table>
```

Running this channel produces five sets of segmented media files, one set for each video output and each audio output.

```
| Video +        |
| Captions (embedded) |
| Video +        |
| Captions (embedded) |
| Video +        |
| Captions (embedded) |
| Audio          |
| Audio          |
```

In outputs that have manifests, it also produces one master manifest and five variant manifests.

**ABR asset with captions in sidecars**

In this example, each captions asset (one for each language) is in its own output.

```
<table>
<thead>
<tr>
<th>Output Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output V</td>
</tr>
<tr>
<td>Output V</td>
</tr>
<tr>
<td>Output V</td>
</tr>
<tr>
<td>Output A</td>
</tr>
<tr>
<td>Output A</td>
</tr>
<tr>
<td>Output C</td>
</tr>
<tr>
<td>Output C</td>
</tr>
</tbody>
</table>
```

Running this channel produces seven sets of segmented media files.

```
| Video         |
| Video         |
| Video         |
| Audio         |
| Audio         |
```

In outputs that have manifests, it also produces one manifest file and seven variant manifests.

**Setting up the downstream system**

You must set up the device or application that will be downstream of MediaLive. The downstream system is different for different outputs.

The output from MediaLive is considered input to this downstream system. You must set up this downstream input now, because when you create the MediaLive channel you need the location of that input.

**Setting up the downstream system for an Archive output**

The downstream system is always an Amazon S3 bucket.

If you decided to set up the channel as a **standard channel** (p. 58), then in Amazon S3, create two buckets, one for each MediaLive channel pipeline.

If you decided to set up a single-pipeline channel, create one bucket.

Make sure that the bucket name doesn't use dot notation. For example, mycompany-videos is acceptable but mycompany.videos isn't. Make a note of the full path of the buckets.
Setting up the downstream system for an HLS output

If you want to deliver HLS output, you must decide if you want to create an HLS output group or a MediaPackage output group. If you choose to create an HLS output group, you must set up the downstream system.

Choosing between HLS and MediaPackage output groups

For most output destinations, you create an HLS output in order to deliver HLS output. But if your destination is a channel in AWS Elemental MediaPackage, you can choose to create an HLS output or a MediaPackage output. There are differences in the setup of each type:

• In the section called "Identifying Resiliency Requirements" (p. 58), you decided whether or not to set up the channel as a standard channel (p. 58)).

  If you decided to set up a single-pipeline channel, then you should create an HLS output group because you will be able to control the behavior when output is lost (p. 102).

  If you decided to set up the channel as a standard channel, you can create either an HLS output or a MediaPackage output.

• 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. For an HLS output, the two channels can be in different Regions (although we recommend that they are in the same Region).

• In a MediaPackage output, the output is always a live stream, not a VOD stream. In an HLS output, you can choose whether to create a live or a VOD stream.

• In a MediaPackage output, there are some restrictions on setting up ID3 metadata. For details, see the section called “ID3 Metadata” (p. 255).

To set up a MediaPackage output, see the section called "For a MediaPackage output" (p. 67).

Setting up for an HLS output

In your downstream system, set up two inputs (for a standard channel (p. 58)), or set up one input (for a single-pipeline channel).

<table>
<thead>
<tr>
<th>Downstream System</th>
<th>Setup</th>
</tr>
</thead>
<tbody>
<tr>
<td>A CDN that uses HTTP (or HTTPS) PUT</td>
<td>Set up the CDN so that the CDN expects MediaLive output at one or two inputs. Make a note of the input addresses, and of the input user name and input password, if applicable.</td>
</tr>
<tr>
<td>A CDN that uses HTTP or HTTPS WebDAV</td>
<td>Create one AWS Elemental MediaPackage channel. See Creating a Channel in the AWS Elemental MediaPackage User Guide.</td>
</tr>
<tr>
<td>An Akamai CDN (this always uses HTTP or HTTPS)</td>
<td>Then view the details of the MediaPackage channel. See Viewing Channel Details in the AWS Elemental MediaPackage User Guide. Make a note of the input URLs, the input user names, and the input passwords.</td>
</tr>
<tr>
<td>AWS Elemental MediaPackage, serving as an origin server that a CDN such as Amazon CloudFront can pull from</td>
<td></td>
</tr>
</tbody>
</table>

66
Downstream System

An AWS Elemental MediaStore container, serving as an origin server that a CDN such as Amazon CloudFront can pull from.

An Amazon S3 bucket, serving as an origin server that a CDN such as Amazon CloudFront can pull from.

Setup

Create one or two containers. Make a note of the full paths of the containers and the data plane values of the containers.

Create one or two buckets.

Make sure that the bucket name doesn't use dot notation. For example, mycompany-videos is acceptable but mycompany.videos isn't. Make a note of the full paths of the buckets.

Setting up the downstream system for a MediaPackage output

A MediaPackage output is an HLS output where the destination is always a channel in AWS Elemental MediaPackage. AWS Elemental MediaPackage is typically serving as an origin server that a CDN such as Amazon CloudFront can pull from.

For information on the differences between HLS and MediaPackage outputs, see the section called “Choosing between HLS and MediaPackage output groups” (p. 66).

To set up AWS Elemental MediaPackage

1. Make sure that the AWS Elemental MediaPackage channel and the AWS Elemental MediaLive channel are in the same AWS Region.
2. In AWS Elemental MediaPackage, create one channel. See Creating a Channel in the AWS Elemental MediaPackage User Guide.
3. View the details of the MediaPackage channel. See Viewing Channel Details in the AWS Elemental MediaPackage User Guide. Make a note of the channel ID. The channel ID is case-sensitive.

Setting up the downstream system for a Frame Capture output

The downstream system is always an Amazon S3 bucket.

If you decided to set up the channel as a standard channel (p. 58), then in Amazon S3, create two buckets, one for each MediaLive channel pipeline.

If you decided to set up a single-pipeline channel, create one bucket.

Make sure that the bucket name doesn't use dot notation. For example, mycompany-videos is acceptable but mycompany.videos isn't. Make a note of the full path of the buckets.

Setting up the downstream system for a Microsoft Smooth output

The downstream system is always a CDN that uses HTTP (or HTTPS) PUT. Typically, the downstream system is a Microsoft IIS server.
If you decided to set up the channel as a **standard channel** (p. 58), then set up the CDN so that the CDN expects MediaLive output at two inputs.

If you decided to set up a single-pipeline channel, then set up the CDN to expect output at only one input.

Make a note of the input addresses, and of the input user name and input password, if applicable.

**Setting up the downstream system for an RTMP or RTMPS output**

The downstream system is always a server that uses RTMP or RTMPS. The server might be a CDN.

If you decided to set up the channel as a **standard channel** (p. 58), set up the RTMP server so that it expects MediaLive output at two inputs.

If you decided to set up a single-pipeline channel, set up the server to expect output at only one input.

Perform the necessary setup on the RTMP server to obtain the following information for each input:

- URL for the output to send to.
- Port number.
- Application name. (The application name might be identical for both inputs.)
- Stream name.

Make a note of the URLs and port numbers.

The URL might include a path portion in the format `<string>/<string>`. In this case, the first string is the application name and the second string is the stream name. In addition, the RTMP server might refer to the stream name as the application instance or the stream key.

**Setting up the downstream system for a UDP output**

The downstream system is an address that can communicate over UDP or RTP.

If you decided to set up the channel as a **standard channel** (p. 58), set up the UDP destination so that it expects MediaLive output at two inputs.

If you decided to set up a single-pipeline channel, set up the destination to expect output at only one input.

Perform the necessary setup at the UDP destination to obtain the following information for each input:

- URL for the output to send to
- Port number

Make a note of the URLs and port numbers.

**Next steps**

Now that you have set up the upstream and downstream systems and have created the inputs that you need, you are ready to create the channel itself. See the section called “Creating a channel from scratch” (p. 69).
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. 62) 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 (p. 69)
- Creating a Channel from a Template or by Cloning (p. 118)
- Editing and Deleting a Channel (p. 121)
- Update the channel class—pipeline resiliency (p. 122)
- Viewing a Channel Configuration (p. 124)

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 channel (p. 57) 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.
Getting ready

Gather the information you noted when you set up the upstream system.

- The name of the input.
- The assets to extract from the source stream.
- The output groups to create.

The name of the input

Collect the name of the input that you created for this channel. If you have not yet created this input, do so now (p. 126). You will need the name of this input when you create the input attachment in order to associate the input with the channel.

The assets to extract

Obtain information about the assets to extract from the source stream. You identified these assets when you assessed the source content (p. 57). You will need this information to create the selectors, starting in the section called “Input settings—Video selector” (p. 78).

- The video asset to extract from the source stream. You need to identify the asset only when the input might contain more than one asset:
  - HLS input – You can let MediaLive extract the highest bandwidth. Or to extract a lower bandwidth asset, obtain its bandwidth.
  - MediaConnect input – For an MPTS, obtain the name of the program to extract.
  - RTP input – For an MPTS, obtain the name of the program to extract.
  - The audio asset or assets to extract from the source stream, if the source stream contains more than one audio asset. For example, AAC in French and Spanish.
  - The captions assets to extract from the source stream, if the source stream contains more than one captions asset. For example, embedded captions in French, Spanish, English, and Italian.
The output groups to create

Obtain information about the output groups to create, including the outputs in each output group and the encodes in each output. You will need this information when you create the output groups (p. 80).

For example:

- One HLS output group that is an ABR stack that contains:
  - Two video outputs, one for a high-resolution video encode, and one for a low-resolution video encode
  - Two audio outputs, one for an AAC French, one for AAC Spanish
  - Four captions outputs, one each for WebVTT captions encodes in French, Spanish, English, and Italian. WebVTT is a sidecar-style captions format, so each captions encode goes in its own output.

  For an illustration of the contents of this output group, see the section called “HLS destination examples” (p. 95).

- One Archive output group that contains one output that contains all the encodes:
  - One high-resolution video encode
  - Two AAC audio encodes
  - Four DVB-Sub captions encodes. DVB-Sub is a object-style captions format, so all the captions encodes go in the same output as the video and audio.

  For an illustration of the contents of this output group, see the section called “Examples” (p. 83).

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

2. Before creating a channel, make sure that you have created the input (p. 126) 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. 72).
   - For information about the Channel template section, see the section called “Creating a Channel from a Template or by Cloning” (p. 118).
   - In Channel class, choose the class. See the section called “Channel class” (p. 72).
   - In Input specifications, complete the fields to match your input. See the section called “Input specifications settings” (p. 73).
   - 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. 319).
6. When ready, go to the next step (p. 73).
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 role 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. 243), 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.

We strongly recommend that you set up all channels in as standard channels.

**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 values in these fields are used to calculate the charges that you will incur on the input side. The values also ensure that MediaLive allocates sufficient processing resources when you run this channel. The fields are the following:

- Input codec
- Input resolution
- Maximum input bitrate

All the fields provide options that cover ranges, with the lowest range shown first and the highest shown last. Lower ranges imply lower processing requirements, and higher ranges imply higher requirements.

For each field, choose an option that meets or exceeds the requirements of your input. If your plan is for your channel to have more than one input, choose the option that meets or exceeds the most demanding of your inputs.

If you don’t choose the correct option, MediaLive might not allocate sufficient processing resources. If you aren’t sure about the processing requirements of your input, 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. Even with codecs, this advice applies. For example, if you aren’t sure if your input is AVC (H.264) or HEVC (H.265), then choose HEVC.

MediaLive uses these values for billing and resource allocation purposes: you pay for the option that you specify. For example, if you specify HD but the input is actually SD, you will be charged for HD.

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**

You must specify the input or inputs to attach to the channel, and then configure how AWS Elemental MediaLive will handle this input.

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. 262). There are specific rules about the number and type (push versus pull, for example) of inputs that you can attach to one channel.

**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. You should have identified this input in the section called “Getting ready” (p. 70). After you choose the input, information about the input appears.

To review this information, see the following sections:

- the section called “Channel input—HLS pull input” (p. 74)
- the section called “Channel input—MediaConnect push input” (p. 74)
- the section called “Channel input—MP4 pull input” (p. 75)
- the section called “Channel input—RTMP push input” (p. 75)
- the section called “Channel input—RTMP pull input” (p. 76)
- the section called “Channel input—RTP push input” (p. 76)

3. For **Attachment name**, enter a name for the attachment. The default name is the name of the input itself. You will need to change this default if you have already attached this input to the channel because names must be unique in the channel.

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. 79).

**Topics**

- Channel input—HLS pull input (p. 74)
- Channel input—MediaConnect push input (p. 74)
- Channel input—MP4 pull input (p. 75)
- Channel input—RTMP push input (p. 75)
- Channel input—RTMP pull input (p. 76)
- Channel input—RTP push input (p. 76)

**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/newschannel/anytownusa.m3u8 and https://203.0.113.54/newschannel/anytownusa.m3u8

Or, for an AWS Elemental MediaStore pull:

mediastoressl://eri39n.data.mediasstore.us-west-2.amazonaws.com/premium/canada/mlaw.m3u8 and mediastoressl://eri39n.data.mediasstore.us-west-2.amazonaws.com/redundant/premium/canada/mlaw.m3u8

**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

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/fillervideos/oceanwaves.mp4
  https://203.0.113.52/fillervideos/oceanwaves.mp4

• For a file that is stored on Amazon S3, the location is the bucket name and object for the file. For example:

  s3://fillervideos/main/oceanwaves.mp4
  s3://fillervideos/redundant/oceanwaves.mp4

With MediaLive, the Amazon S3 bucket name mustn't use dot notation. For example, mycompany-videos is acceptable but mycompany.videos isn't.

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://203.0.113.111:1935/movies/classic
rtmp://203.0.113.22:1935/movies/classic

For example, for an RTMP VPC push input:
rtmp://192.0.2.131:1935

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—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://203.0.113.20:1935/movies/classic
rtmp://203.0.113.88:1935/movies/classic

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://203.59.21.50:0:5000
rtp://203.59.21.131:5000
For example, for an RTP VPC input:

```
rtp://10.99.39.87.30.100:5000
```

```
rtp://192.0.2.123:5000
```

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:
   - 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. 262).

3. When ready, go to the **next step** (p. 79).

#### Topics

- Input settings—Network input settings (p. 77)
- Input settings—Other settings (p. 77)
- Input settings—Video selector (p. 78)
- Input settings—Audio selectors (p. 79)
- Input settings—Caption selectors (p. 79)

### Input settings—Network input settings

Complete this section only if the input is HLS.

### 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.

- **Selector settings**: This field lets you identify the video to ingest. Complete this field or ignore it, depending on the input type:

<table>
<thead>
<tr>
<th>Input type</th>
<th>Method of extracting video</th>
</tr>
</thead>
<tbody>
<tr>
<td>HLS</td>
<td>Don’t complete <strong>Selector settings</strong>. These extraction methods don’t apply to HLS inputs. By default, MediaLive extracts the video asset with the highest bandwidth. You can complete the <strong>Bandwidth</strong> field (in input settings – Network input settings). MediaLive extracts the highest bandwidth video that is below this limit.</td>
</tr>
<tr>
<td>MediaConnect</td>
<td>Complete <strong>Selector settings</strong> if the input contains an MPTS, 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 <strong>Selector settings</strong>.</td>
</tr>
<tr>
<td>MP4</td>
<td>The input contains only one video asset. MediaLive extracts that video. There is no need to complete <strong>Selector settings</strong>.</td>
</tr>
<tr>
<td>RTMP</td>
<td>The input contains only one video asset. MediaLive extracts that video. There is no need to complete <strong>Selector settings</strong>.</td>
</tr>
<tr>
<td>RTP</td>
<td>Complete <strong>Selector settings</strong> if the input contains an MPTS, 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 <strong>Selector settings</strong>.</td>
</tr>
</tbody>
</table>

Keep in mind that there is no button to add more video selectors because you can extract only one video asset from the input.

- **Color space** and **Color space usage**: These fields let you configure the optional color space feature. For more information, see the section called “Color Space” (p. 246).
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 to extract. Typically, you identify different languages from the input, but you could also extract different audio codecs (such as AAC and Dolby).

For each audio that you want to extract, choose Add audio selector. Complete the fields that appear to identify the location of the audio and to specify optional handling of the audio.

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. 224), specifically the section called “Step 1: Create Captions Selectors in the Input” (p. 229).

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. 80).

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. 299).

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. 299).

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. 299).

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.

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. 297).

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. 321).

Logging

Optional feature. You can enable logging of activity on this individual channel. For detailed information about this feature, see the section called “Monitoring a Channel Using CloudWatch Logs” (p. 205).

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 that you identified when you planned the channel (p. 57). AWS Elemental MediaLive supports different output types. For more information, see the section called “Supported Codecs for Outputs” (p. 341).

**Topics**
- Creating an archive output group (p. 80)
- Creating a frame capture output group (p. 86)
- Creating an HLS output group (p. 88)
- Creating a MediaPackage output group (p. 103)
- Creating a Microsoft Smooth output group (p. 106)
- Creating an RTMP output group (p. 111)
- Creating a UDP output group (p. 114)

Creating an archive output group

Read this section if, when you were planning the channel (p. 57), you determined that you want to include an archive output group.
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. 81).
   - Archive settings – This section contains fields for the output container (p. 85).
   - 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. 81) and the output container (p. 85).
   - Stream settings – This section contains fields for the output streams (p. 85) (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 tvchannel159.
   - 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. 118).

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 – Additional settings – Rollover interval
- Archive outputs – Name modifier
- Archive outputs – Extension

To specify the location for the output

1. Design the destination URL. See the section called “About the path for the output destination” (p. 82) for information on the syntax of the URL.
2. Enter the different portions of the destination in the appropriate fields.
### Step 5: Create output groups

#### About the path for the output destination

The location and file names for an Archive output follow this syntax:

<table>
<thead>
<tr>
<th>Portion of the destination URL</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>protocol, path, base_filename</td>
<td>The two URL fields in the Archive group destinations section. Specify two destinations when the channel is set up as a standard channel, or one destination when it is set up as a single-pipeline channel.</td>
</tr>
<tr>
<td>name_modifier</td>
<td>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. 343).</td>
</tr>
<tr>
<td>extension</td>
<td>The Extension field in the Archive outputs. Always leave the default, m2ts.</td>
</tr>
</tbody>
</table>

For more information, see the examples (p. 83).

**3.** Leave the Credentials section blank in both the Archive group destinations sections. MediaLive has permission to write to the Amazon S3 bucket via the trusted entity. Someone in your organization should have already set up these permissions. For more information, see the section called “Summary Of Requirements for the MediaLive Trusted Entity” (p. 41).

**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 base_filename, the name_modifer, and a sequential counter.

The current file is visible on Amazon S3 only after it has closed.

For more information, see the examples (p. 83).

**About the path for the output destination**

The location and file names for an Archive output follow this syntax:

<table>
<thead>
<tr>
<th>protocol path base_filename name_modifer counter extension</th>
</tr>
</thead>
<tbody>
<tr>
<td>For example: s3ssl://channel59/delivery/program_20171012T033162_000001.m2ts</td>
</tr>
</tbody>
</table>

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 Amazon S3 bucket. |
path | channel59/delivery/ | The path consists of a bucket and folders, always terminated by a slash. With MediaLive, the Amazon S3 bucket name must not use dot notation. For example, mycompany-videos is acceptable but mycompany.videos isn’t. |
base_filename | program | Don't terminate the file name with a slash. |
name_modifier | _20171012T033162 | The modifier is optional for an Archive output. |
delimiter before the counter | . | MediaLive automatically inserts this delimiter. |
counter | 000001 | MediaLive automatically generates this counter. Initially, this is a six-digit number: 000001, 000002, and so on. If the counter reaches 999999, it becomes a seven-digit counter: 0000001, 0000002, and so on. |
dot before the extension | . | MediaLive automatically inserts this dot. |
extension | m2ts | Always m2ts. |

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 Amazon S3 bucket named channel59, and you want to break up the stream into 5-minute chunks.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rollover interval field in Archive settings section</td>
<td>300</td>
</tr>
<tr>
<td>URL in Archive group destination A section</td>
<td>s3ssl://channel59/delivery/program</td>
</tr>
</tbody>
</table>
### Step 5: Create output groups

#### Field | Value
--- | ---
URL in Archive group destination B section | `s3ssl://channel59/backup/program`<br>Using *delivery* and *backup* as folder names is only an example.

Name modifier in Archive outputs section | `_dt#`
For information about identifiers for variable data (such as `$dt$`), see the section called “Reference: Identifiers for Variable Data” (p. 343).

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 `program`, the time that the channel started and a counter (000001, 000002, and so on), and the file name extension. For example:

- The first file will be `program_20171012T033162_000001.m2ts`.
- The second file will be `program_20171012T033162_000002.m2ts`.

Each file will be stored in both `s3ssl://channel59/delivery` and `s3ssl://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 Amazon S3 bucket named `sports/highlights/curling`. 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://sports/delivery/highlights/curling/10312017`<br>In this example, the `10312017` folder is set to match today's date.

URL in Archive group destination B section | `s3ssl://sports/backup/highlights/curling/10312017`<br>Using *delivery* and *backup* as folder names is only an example.

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 sets of files are created for each output. Each file has a file name of 10312017, plus the modifier, the sequence counter, and the file name extension. For example:

- 10312017_audiogroup1_000001.m2ts, 10312017_audiogroup2_000001.m2ts, and 10312017_audiogroup3_000001.m2ts.
- 10312017_audiogroup1_000002.m2ts, 10312017_audiogroup2_000002.m2ts, and 10312017_audiogroup3_000002.m2ts.

Each file will be stored in both s3ssl://sports/delivery/highlights/curling and s3ssl://sports/backup/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

By default, the output is set up with one video encode and one audio encode. The output in an Archive group can contain one video encode, one or more audio encodes, and zero or more captions encodes.

**To set up the encodes**

1. Decide if you need to create more encodes for this output, based on the workflow that you planned (p. 62).
2. Choose the appropriate **Add** button or **Delete** button to set up the output with the encodes—video, audio, and captions—that you planned for this output.

For example, in one output you might want one video asset and two audio assets, in another output you might want one captions asset for French captions, and in a third output you might want one captions asset for Spanish captions. For information about why you would set up like this (for example, with an output that contains only one captions asset), see the section called “Examples of channel designs” (p. 63).

3. Complete the fields in each type of encode.

   For information about the fields in each type of encode, see the following sections:
   - the section called “Step 6: Set up video” (p. 116)
   - the section called “Step 7: Set up audio” (p. 117)
   - the section called “Step 8: Set up captions” (p. 118)
Creating a frame capture output group

Follow these steps if, when you were planning the channel (p. 57), you determined that you want to include 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 Amazon S3 bucket. With MediaLive, the Amazon S3 bucket name must not use dot notation. For example, mycompany-videos is acceptable but mycompany.videos isn’t.

Topics

- Procedure to create a frame capture output group (p. 86)
- Frame capture destination (p. 86)
- Settings for the stream (p. 88)

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. 86).
   - Frame capture settings – This section contains a field for the output group name.
   - 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. 86).
   - Stream settings – This section contains fields for the output streams (p. 88) (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. 118).

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 settings – Name modifier

To specify the location and file name for the output

1. Design the destination URL. See the section called “About the path for the output destination” (p. 87) for information on the syntax of the URL.
2. Enter the different portions of the destination in the appropriate fields.

<table>
<thead>
<tr>
<th>Portion of the destination URL</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>protocol, path, base_filename</td>
<td>The two URL fields in the Frame capture group destinations section.</td>
</tr>
<tr>
<td></td>
<td>Specify two destinations when the channel is set up as a standard channel, or one destination when it is set up as a single-pipeline channel.</td>
</tr>
<tr>
<td>name_modifier</td>
<td>The Name modifier field in the Frame capture outputs section.</td>
</tr>
</tbody>
</table>

3. Leave the Credentials section blank in both the Frame capture group destinations sections. MediaLive has permission to write to the Amazon S3 bucket via the trusted entity. Someone in your organization should have already set up these permissions. For more information, see the section called “Summary Of Requirements for the MediaLive Trusted Entity” (p. 41).

About the path for the output destination

The location and file names for a frame capture output follow this syntax:

```
protocol path base_filename name_modifier counter extension
```

For example:

```
s3ssl://sports_thumbnails/curling_20180820_000001.jpg
```

The following table maps each portion in the example to the portion in the syntax.

<table>
<thead>
<tr>
<th>Portion of the destination URL</th>
<th>Example</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>protocol</td>
<td>s3ssl://</td>
<td>The protocol is always s3ssl:// because the destination for a frame capture output is always an Amazon S3 bucket.</td>
</tr>
<tr>
<td>path</td>
<td>sports_thumbnails/</td>
<td>The path consists of a bucket and folders, always terminated by a slash.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>With MediaLive, the Amazon S3 bucket name must not use dot notation. For example, mycompany-videos is acceptable but mycompany.videos isn't.</td>
</tr>
</tbody>
</table>
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. 116).

Creating an HLS output group

Follow these steps if, when you were planning the channel (p. 57), you determined that you want to include an HLS output group.

Topics

- Procedure to create an HLS output group (p. 89)
- Fields for the output destination (p. 89)
- Fields for the HLS container (p. 99)
- Fields for customizing the paths inside the manifests (p. 100)
- Fields for redundant manifests (p. 100)
- Fields for the video, audio, and captions streams (encodes) (p. 100)
- Fields for other HLS features (p. 101)
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 (p. 89).
   • HLS settings – This section contains fields for the destination of the outputs (p. 89), for resiliency (p. 102), and for captions (p. 103).
   • 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. 280).
   • Manifest and segments – This section contains fields for configuring redundant manifests (p. 100), for configuring the manifest contents (p. 101), and for configuring media segments (p. 102).
   • DRM – This section contains fields for configuring encryption of outputs (p. 102).
   • Ad marker – This section contains fields for setting up for SCTE-35 ad avails (p. 103).
   • Captions – This section contains fields for configuring captions (p. 103).
   • ID3 – This section contains fields for setting up for ID3 (p. 103).
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 output destination (p. 89), and for the HLS container (p. 99).
   • Stream settings – This section contains fields for the output streams (p. 100) (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. 118).

Fields for the output destination

You must specify where you want to send the output of the output group. Before you complete the relevant fields, you must speak to the downstream system to discuss your requirements.

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
Step 5: Create output groups

- HLS outputs – Output settings – Segment modifier

Topics
- Step 1: Obtain information (p. 90)
- Step 2: Design output names and destination paths (p. 91)
- Step 3: Set up the output names and destination paths (p. 91)
- About the protocol (p. 93)
- About the path for output destinations (p. 94)
- HLS destination examples (p. 95)

Step 1: Obtain information

Before you start to configure an HLS output group, you must obtain information from the downstream system. There is a different procedure for MediaPackage compared to other downstream systems.

To obtain information from most downstream systems

1. Find out if the downstream system has special connection requirements. These connection fields are grouped in the CDN settings section on the HLS output group page.
2. Find out what protocol the downstream system is using. See the section called “About the protocol” (p. 93).
3. Obtain the protocol/domain/path where the downstream system wants to find the three sets of files (the main manifests, the child manifests, and the media files).

   MediaLive always puts all three sets of files in this one location. It’s not possible to configure MediaLive to put some files in another location.
4. 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. 280).
5. 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 “Redundant manifest fields” (p. 100).
6. Find out if the downstream system has special requirements for the base_filename portions of the destination URLs. Typically, the downstream system doesn’t have special requirements, so you will choose these names in the design step.
7. Find out if the downstream system has special requirements for the name_modifier portions of the destination URLs. The child manifests and media files include this modifier in their file names. This name_modifier distinguishes each output from the other, so it must be unique in each output.

   Typically, the downstream system doesn’t have special requirements, so you will choose these names in the design step.
8. Ask 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.

To obtain information from MediaPackage

- Obtain information about the channel where you will send the output from MediaLive. You need the input URLs, the input user names, and the input passwords.

   You don’t use the channel ID, so you don’t need that ID.
Step 2: Design output names and destination paths

You should take some time to carefully plan the output names and destination paths for the HLS output. Skip this step if the downstream system is MediaPackage.

To design outputs and destinations

1. Read the information in the section called “About the path for output destinations” (p. 94) for general information about the different portions of the output names and destination paths.
2. Design the destination path.

   For the base_filename portion of the destination path, follow these guidelines:
   - For a single-pipeline channel, you need only one base_filename.
   - For a standard channel when you are not implementing redundant manifests (p. 100), you need two base_filenames. The two base_filenames can be identical or different. Before you create different base_filenames, 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. 100).
3. Design the name_modifier portions of the destination path. The child manifests and media files include this modifier in their file names. This name_modifier 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 an underscore, to clearly separate the base_filename from the name_modifier.
   - The name_modifier can include data variables (p. 343).
4. Design the segment_modifiers portion of the destination path. The segment_modifiers 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 000002 might have the name curling_120208_000002. If the output restarts a few minutes later (which causes the segment counter to restart), the new segment 000002 will have the name curling_120209_000002. The new file won't overwrite the file for the original segment 000002. Some downstream systems might prefer this behavior.

Step 3: Set up the output names and destination paths

After you have designed the output names and destination paths, you can set up the HLS output group.

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.

<table>
<thead>
<tr>
<th>Portion of the destination URL</th>
<th>Example</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>protocol</td>
<td>http://</td>
<td>The protocol that you obtained from the downstream system.</td>
</tr>
</tbody>
</table>
### Portion of the destination URL

<table>
<thead>
<tr>
<th>Portion of the destination URL</th>
<th>Example</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>domain</td>
<td>203.0.113.55</td>
<td>The IP address or host/domain that you obtained from the downstream system.</td>
</tr>
<tr>
<td>path</td>
<td>/sports/delivery/</td>
<td>The path that you obtained from the downstream system. Always terminate with a slash.</td>
</tr>
<tr>
<td>base_filename</td>
<td>curling</td>
<td>The name that you and the downstream system agreed on. Don't terminate the base_filename with a slash.</td>
</tr>
</tbody>
</table>

2. In each **HLS group destination** section, complete the **Credentials** section as appropriate for the protocol that you chose.

<table>
<thead>
<tr>
<th>Protocol of the destination URL</th>
<th>How to complete the Credentials</th>
</tr>
</thead>
<tbody>
<tr>
<td>http://</td>
<td>Typically, you don't enter credentials</td>
</tr>
<tr>
<td>https://</td>
<td>In each <strong>Credentials</strong> section, enter a user name and a password that is known to 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. 31).</td>
</tr>
<tr>
<td>s3://</td>
<td>Leave both the <strong>Credentials</strong> sections blank. MediaLive has permission to write to the Amazon S3 bucket via the trusted entity. Someone in your organization should have already set up these permissions. For more information, see the section called “Step 1: Determine Requirements” (p. 41).</td>
</tr>
<tr>
<td>s3ssl://</td>
<td>Leave both the <strong>Credentials</strong> sections blank. MediaLive has permission to write to the Amazon S3 bucket via the trusted entity. Someone in your organization should have already set up these permissions. For more information, see the section called “Step 1: Determine Requirements” (p. 41).</td>
</tr>
<tr>
<td>mediastoressl://</td>
<td>Leave both the <strong>Credentials</strong> sections blank. MediaLive has permission to write to the Amazon S3 bucket via the trusted entity. Someone in your organization should have already set up these permissions. For more information, see the section called “Step 1: Determine Requirements” (p. 41).</td>
</tr>
</tbody>
</table>

3. In the **HLS outputs** section, complete the **Name modifier** for each output, according to your design.

4. (Optional) In the **HLS outputs** section, complete the **Segment modifier** for outputs, according to your design.

5. In the **CDN** settings section, choose the option that applies to the downstream system.

6. If the downstream system gave you values to configure the connection, enter those values in the fields in the **CDN** settings section.
To set the destination for MediaPackage

1. Complete the HLS group destination A section:
   - For URL, enter the URL of the input in the MediaPackage channel. For example:
     https://62e3c93793c034c.mediapackage.us-west-2.amazonaws.com/in/v1/9378dje8/channel
   - For Credentials, 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. 31).

2. If you are setting up a standard channel, in HLS group destination B, enter the information for the other input. If you are setting up a single channel, leave this section empty.

3. In the Location section, don't change any of the fields.

4. In the HLS outputs section, complete Name modifier as follows:
   - 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 an underscore, to clearly separate the base_filename from the name modifier.
   - The name_modifier can include data variables (p. 343).

5. (Optional) In the HLS outputs section, complete Segment modifier as follows:
   - The segment_modifier is optional, but we strongly recommend that you 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 000002 might have the name curling_120028_000002. If the output restarts a few minutes later (which causes the segment counter to restart), the new segment 000002 will have the name curling_120039_000002. The new file won't overwrite the file for the original segment 000002. Some downstream systems might prefer this behavior.


About the protocol

This table describes the different protocols that MediaLive supports for downstream system. Each row in the table describes a different protocol.

<table>
<thead>
<tr>
<th>Type of downstream system</th>
<th>CDN field</th>
<th>Protocol for destination URL</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>A downstream system that uses HTTP or HTTPS PUT</td>
<td>Hls basic put</td>
<td>http://</td>
<td>If you send to an Amazon S3 bucket, the bucket name must not use dot notation. For example, mycompany-videos is acceptable but</td>
</tr>
<tr>
<td>Amazon S3 bucket</td>
<td>Hls basic put</td>
<td>s3://</td>
<td></td>
</tr>
</tbody>
</table>
Step 5: Create output groups

<table>
<thead>
<tr>
<th>Type of downstream system</th>
<th>CDN field</th>
<th>Protocol for destination URL</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>MediaStore container</td>
<td>Hls media store</td>
<td>mediastoresssl://</td>
<td></td>
</tr>
<tr>
<td>Akamai CDN</td>
<td>Hls Akamai</td>
<td>http://</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>https://</td>
<td></td>
</tr>
<tr>
<td>AWS Elemental MediaPackage</td>
<td>Hls webdav</td>
<td>http://</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>https://</td>
<td></td>
</tr>
<tr>
<td>A downstream system that uses WebDAV</td>
<td>Hls webdav</td>
<td>http://</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>https://</td>
<td></td>
</tr>
</tbody>
</table>

About the path for output destinations

HLS outputs consist of three sets of files: the main manifest, the child manifests, and the media files (segments).

MediaLive sends all these sets of files to the same destination (protocol, domain, path). The file names are different for each.

Topics

- The syntax for the URLs of HLS files (p. 94)
- How MediaLive constructs these URLs (p. 95)

The syntax for the URLs of HLS files

The full URLs have the following syntax.

The syntax for the URL for main manifest files

The syntax is the following:

```
protocol domain path base_filename extension
```

For example, the URL for the master manifest for the curling output might be:

```
http:// 203.0.113.55/sports/delivery/curling.m3u8
```

The syntax for the URL for child manifest files

The syntax is the following:

```
protocol domain path base_filename modifier extension
```

For example, the URL for the child manifest for the high-resolution renditions of the curling output might be:
The syntax for the URL for the media files (segments)

The syntax is the following:

```
protocol domain path base_filename modifier optional_segment_modifier counter extension
```

For example, the URL for the file for the 230th segment might be:

```
http://203.0.113.55/sports/delivery/curling_high_00230.ts
```

Note that the URLs for all the files are identical to each other, up to and including the `base_filename`.

How MediaLive constructs these URLs

These URLs are constructed as follows:

- You specify the protocol, domain, path, `base_filename`, modifier, and `segment Modifier`.
- MediaLive inserts the underscore before the counter.
- MediaLive generates the counter, which is always six digits starting at 000002.
- 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

HLS destination examples

Topics

- Example for an HTTP or HTTPS server (p. 95)
- Akamai example (p. 96)
- MediaPackage example (p. 98)

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.

Assume that you want to stream the curling game and to create three outputs: high, medium, and low bitrate.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CDN settings in HLS settings section</strong></td>
<td>HLS basic put</td>
</tr>
<tr>
<td><strong>URL in HLS group destination A section</strong></td>
<td>For example:</td>
</tr>
<tr>
<td><strong>Credentials in HLS group destination A section</strong></td>
<td>The protocol is HTTPS, which typically means that you must authenticate your request.</td>
</tr>
</tbody>
</table>
## Step 5: Create output groups

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Enter the user name provided by the downstream system.</strong></td>
<td>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 &quot;AWS Systems Manager Parameter Store&quot; (p. 31).</td>
</tr>
<tr>
<td><strong>URL in HLS group destination B section</strong></td>
<td>For example:</td>
</tr>
<tr>
<td><strong>Credentials in HLS group destination B section</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Name modifier in HLS outputs section</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Directory Structure and Segments Per Subdirectory in Location section</strong></td>
<td>Assume that the downstream system doesn't use these fields.</td>
</tr>
</tbody>
</table>

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_000001.ts, curling_high_000002.ts, curling_high_000003.ts, and so on
  - curling_medium_000001.ts, curling_medium_000002.ts, curling_medium_000003.ts, and so on
  - curling_low_000001.ts, curling_low_000002.ts, curling_low_000003.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.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CDN settings in HLS settings section</strong></td>
<td>Hls akamai</td>
</tr>
<tr>
<td><strong>Change the other CDN fields according to the instructions from Akamai.</strong></td>
<td></td>
</tr>
<tr>
<td><strong>URL in HLS group destination A section</strong></td>
<td>For example:</td>
</tr>
</tbody>
</table>
### Step 5: Create output groups

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong><a href="http://p-ep50002.i.akamaientrypoint.net/50002/curling">http://p-ep50002.i.akamaientrypoint.net/50002/curling</a></strong></td>
<td></td>
</tr>
<tr>
<td>Mapping this URL to the Akamai terminology:</td>
<td></td>
</tr>
<tr>
<td>• <em>p-ep</em> stands for primary entry point</td>
<td></td>
</tr>
<tr>
<td>• <em><a href="http://p-ep50002.i.akamaientrypoint.net">http://p-ep50002.i.akamaientrypoint.net</a></em> is the hostname</td>
<td></td>
</tr>
<tr>
<td>• <em>5002</em> (in both places) is the stream ID</td>
<td></td>
</tr>
<tr>
<td>• <em>curling</em> is the event name</td>
<td></td>
</tr>
</tbody>
</table>

| **Credentials in HLS group destination A section** | If your agreement with Akamai is to deliver over HTTPS, 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. 31). |       |
| **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.net* 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**
Step 5: Create output groups

- TS files for each output:
  - curling_high_000001.ts, curling_high_000002.ts, curling_high_000003.ts, and so on
  - curling_medium_000001.ts, curling_medium_000002.ts, curling_medium_000003.ts, and so on
  - curling_low_000001.ts, curling_low_000002.ts, curling_low_000003.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.

MediaPackage 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.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CDN settings in HLS settings section</strong></td>
<td>hls webdav</td>
</tr>
<tr>
<td></td>
<td>This is the type of connection that MediaPackage uses. Change the other CDN fields that appear, or keep the defaults.</td>
</tr>
<tr>
<td><strong>URL in HLS group destination A section</strong></td>
<td>For example, <a href="https://62e3c93793c034c.mediapackage.us-west-2.amazonaws.com/in/v1/9378dje8/channel">https://62e3c93793c034c.mediapackage.us-west-2.amazonaws.com/in/v1/9378dje8/channel</a>.</td>
</tr>
<tr>
<td></td>
<td>The URLs are the Input URLs from the channel in AWS Elemental MediaPackage. As discussed in the section called “Setting up the downstream system” (p. 65), the input in MediaPackage is identical to the output from AWS Elemental MediaLive. This input URL must already exist in MediaPackage.</td>
</tr>
<tr>
<td></td>
<td>Note that in MediaPackage, URLs always end in channel, so the base file name in MediaLive must be channel.</td>
</tr>
<tr>
<td><strong>Credentials in HLS group destination A section</strong></td>
<td>MediaPackage accepts only authenticated requests, so you must enter a user name and a 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. 31).</td>
</tr>
<tr>
<td><strong>URL in HLS group destination B section</strong></td>
<td>For example, <a href="https://60dei849783734c.mediapackage.us-west-2.amazonaws.com/in/v1/6da5ba717b357a/channel">https://60dei849783734c.mediapackage.us-west-2.amazonaws.com/in/v1/6da5ba717b357a/channel</a>.</td>
</tr>
</tbody>
</table>
Field | Value
--- | ---
**Credentials** in **HLS group destination B** section | Enter a user name and password for the URL for destination B. The credentials are probably the same for both URLs, but they might not be.

**Name modifier** in **HLS outputs** section | Choose **Add output** twice: two more **Output** lines are added to this section, for a total of three lines. In each line, enter a modifier: **_high**, **_medium**, and **_low**.

**Directory Structure** and **Segments Per Subdirectory** in **Location** section | MediaPackage 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_000001.ts`, `channel_high_000002.ts`, `channel_high_000003.ts`, and so on
  - `channel_medium_000001.ts`, `channel_medium_000002.ts`, `channel_medium_000003.ts`, and so on
  - `channel_low_000001.ts`, `channel_low_000002.ts`, `channel_low_000003.ts`, and so on

The files will be published to both URL inputs on MediaPackage.

**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. 101) 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.
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 Rendition Groups for HLS” (p. 210).

**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. 280).

**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. 285).

**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
To create more encodes

1. Decide if you need to create more encodes for this output, based on the workflow that you planned (p. 62). By default, each output is set up with one video encode and one audio encode.

2. In Stream settings, choose the appropriate Add button or Delete button to set up the output with the encodes—video, audio, and captions.

   For example, in one output you might want one video asset and two audio assets, in another output you might want one captions asset for French captions, and in a third output you might want one captions asset for Spanish captions. For information about why you would set up like this (for example, with an output that contains only one captions asset), see the section called “Examples of channel designs” (p. 63).

3. Complete the fields in each type of encode.

   For information about the fields in each type of encode, see the following sections:
   - the section called “Step 6: Set up video” (p. 116)
   - the section called “Step 7: Set up audio” (p. 117)
   - the section called “Step 8: Set up captions” (p. 118)

Fields for other HLS features

Topics
- Fields for contents of manifests (p. 101)
- Fields for segments (p. 102)
- Fields for resiliency (p. 102)
- Fields for DRM (p. 102)
- Fields for SCTE-35 ad avails (p. 103)
- Fields for captions (p. 103)
- Fields for ID3 metadata (p. 103)

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
- 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. 72):

- 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. 103) 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. 299) and specifically the section called “Enabling Decoration – HLS” (p. 306).

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. 236).

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. 257).

Creating a MediaPackage output group

Follow these steps if, when you were planning the channel (p. 57), you determined that you want to include a MediaPackage output group.

**Topics**

- Procedure to create a MediaPackage output group (p. 103)
- Streams section (p. 104)
- Result of this procedure (p. 106)

**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. You made a note of this ID when you set up the channel on AWS Elemental MediaPackage.

Enter the channel ID, not the channel ARN. A channel ID is a name you assign, for example, sports_delivery_from_medialive.

4. (Optional) In the MediaPackage settings section, for Name, enter a name for the output group.

5. 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.

6. Choose the first Settings link to view the sections for the first output. The section contains fields for the output streams (p. 100) (the video, audio, and captions).

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. 118).

Streams section

In Stream settings, decide if you need to create more encodes for this output, based on the workflow that you planned (p. 62). By default, each output is set up with one video encode and one audio encode. Choose the appropriate Add button or Delete button to set up the output with the encodes—video, audio, and captions—that you planned for this output.

For example, in one output you might want one video asset and two audio assets, in another output you might want one captions asset for French captions, and in a third output you might want one captions asset for Spanish captions. For information about why you would set up like this (for example, with an output that contains only one captions asset), see the section called “Examples of channel designs” (p. 63).

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. 116)
- the section called “Step 7: Set up audio” (p. 117)
• the section called “Step 8: Set up captions” (p. 118)

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 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 name_modifier 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. 299).

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. 255).

Creating a Microsoft Smooth output group

Follow these steps if, when you were planning the channel (p. 57), you determined that you want to include a Microsoft Smooth output group.

Topics

• Procedure to create a Microsoft Smooth output group (p. 107)
• Fields for the output destination (p. 107)
• Fields for the container (p. 109)
• Fields for the encodes (p. 109)
• Fields for other Microsoft Smooth features (p. 110)

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. 107).
   - Microsoft Smooth settings – This section contains fields for the container (p. 109), the connection to the downstream system (p. 107), and resiliency (p. 110).
   - 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. 107) and the container (p. 109).
   - Timecode configuration – This section contains fields for the timecode (p. 110) in the outputs.
   - Sparse track – This section contains fields for the container (p. 109).
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. 107), and the container (p. 109).
   - Stream settings – This section contains fields for the output streams (p. 109) (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. 118).

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

To specify the path and connection to the downstream system

1. Obtain the protocol/domain/path for the publishing point of the downstream system.
2. Find out if the downstream system has special connection requirements. These connection fields are grouped in the Microsoft Smooth settings section, in the General configuration section.
3. Find out if the downstream system is using the http:// or https:// protocol. See the section called “About the protocol” (p. 93).
4. Design the publishing point. (On the console the publishing points are represented by the destination URL fields.) See the section called “About the path for the output destinations” (p. 108) for information on the syntax of the publishing point.
5. In the URL fields in the Microsoft Smooth group destinations section, enter the protocol, domain, and path. For example:

```
https://203.0.113.18/sports/
```

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.
6. Complete the Credentials section, if the server requires that you authenticate with a user name and password. Make sure that you have obtained the user name and password from 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. 31).
7. If the downstream system gave you values to configure the connection, enter those values in the General configuration section on the Microsoft Smooth group page.

About the path for the output destinations

The publishing point for a Microsoft Smooth output follows this syntax:

```
protocol domain path event_ID
```

For example:

```
https://203.0.113.18/sports/Events(1585232182)
```

The following table maps each portion in the example to the portion in the syntax.

<table>
<thead>
<tr>
<th>Portion of the destination URL</th>
<th>Example</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>protocol</td>
<td>https://</td>
<td>The protocol is either http:// or https://</td>
</tr>
<tr>
<td>domain</td>
<td>203.0.113.18/</td>
<td>The IP address or host/domain that you obtained from the downstream system.</td>
</tr>
<tr>
<td>path</td>
<td>sports/</td>
<td>The path that you obtained from the downstream system. Always terminated by a slash.</td>
</tr>
</tbody>
</table>
### The event ID

The following fields relate to the event ID:

**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.

### The full path for outputs

The full path for the output consists of the publishing point path, plus stream information. You can see the stream information when you look at the MediaLive logs for the output.

MediaLive generates the stream ID. It assigns a unique number to the stream, starting from 0. For example:

`/Streams(stream0)`

### 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 behaviour**

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 encodes in each output.

- **Microsoft Smooth output – Stream settings** section

### To create more encodes

1. Decide if you need to create more encodes for this output, based on the workflow that you planned (p. 62). By default, each output is set up with one video encode and one audio encode.
2. In **Stream settings**, choose the appropriate **Add** button or **Delete** button to set up the output with the encodes—video, audio, and captions.

   For example, in one output you might want one video asset and two audio assets, in another output you might want one captions asset for French captions, and in a third output you might want one captions asset for Spanish captions. For information about why you would set up like this (for example, with an output that contains only one captions asset), see the section called “Examples of channel designs” (p. 63).

3. Complete the fields in each type of encode.

   For information about the fields in each type of encode, see the following sections:

   - the section called “Step 6: Set up video” (p. 116)
   - the section called “Step 7: Set up audio” (p. 117)
   - the section called “Step 8: Set up captions” (p. 118)

**Fields for other Microsoft Smooth features**

**Topics**

- Fields for resiliency (p. 110)
- Fields for timecode (p. 110)
- Fields for SCTE-35 (p. 110)

**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. 299) and specifically the section called “Enabling Decoration – Microsoft Smooth” (p. 306).
Creating an RTMP output group

Follow these steps if, when you were planning the channel (p. 57), you determined that you want to include an RTMP output group.

**Topics**

- Procedure to create an RTMP output group (p. 111)
- Fields for the output destination (p. 111)
- Fields for the RTMP connection (p. 113)
- Fields for the video, Audio, and captions streams (encodes) (p. 114)
- Other fields (p. 114)

**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. 113), for resiliency (p. 114), and for captions (p. 114).
   - 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. 111).
   - Output settings – This section contains fields for the connection configuration (p. 113).
   - Stream settings – This section contains fields for the output streams (p. 114) (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. 118).

**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. Design the destination URL. See the section called “About the path for the output destinations” (p. 112) for information on the syntax of the URL.
2. Enter the different portions of the destination in the appropriate fields.

<table>
<thead>
<tr>
<th>Portion of the destination URL</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>protocol, path, port, application name</td>
<td>The two URL fields in the <strong>RTMP destination</strong> section. (Note that these fields are on the <strong>Output</strong> page, not the <strong>Output group</strong> page.) Specify two destinations when the channel is set up as a <strong>standard channel</strong>, or one destination when it is set up as a single-pipeline channel. See the next step for information about credentials. For example: <strong>rtmp://203.0.113.28:80/live</strong></td>
</tr>
<tr>
<td>stream name</td>
<td>The two <strong>Stream name</strong> fields in the <strong>RTMP outputs</strong> section. For example: <strong>MyStream</strong></td>
</tr>
</tbody>
</table>

3. Complete the **Credentials** section, if the server requires that you authenticate with a user name and password. Make sure that you have obtained the user name and password from the RTMP 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. 31).

Note that these credentials relate to user authentication, not to the protocol. User authentication is about whether the RTMP server will accept your request. The protocol is about whether the request is sent over a secure connection.

**About the path for the output destinations**

The location for an RTMP output follows this syntax:

```
protocol path port application_name application_instance
```

For example:

```
rtmp://203.0.113.28:1935/live/MyStream
```

The following table maps each portion in the example to the portion in the syntax.

<table>
<thead>
<tr>
<th>Portion of the destination URL</th>
<th>Example</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>protocol</td>
<td>rtmp://</td>
<td>The protocol is always <strong>rtmp://</strong> or <strong>rtmps://</strong></td>
</tr>
<tr>
<td>path</td>
<td>203.0.113.28</td>
<td>The IP address or host/domain that you obtained from the downstream system.</td>
</tr>
<tr>
<td>port</td>
<td>:80</td>
<td>The port number that you obtained from the downstream system.</td>
</tr>
</tbody>
</table>
### 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 encodes in the output.

- **Stream settings** section

By default, the output is set up with one video encode and one audio encode. The output in an RTMP group can contain one video encode, zero or one audio encodes, and zero or one captions encodes.

**To set up the encodes**

1. Decide if you need to create more encodes for this output, based on the workflow that you planned (p. 62).
2. Choose the appropriate [Add](#) button or [Delete](#) button to set up the output with the encodes—video, audio, and captions—that you planned for this output.
3. Complete the fields in each type of encode.

   For information about the fields in each type of encode, see the following sections:
   
   - the section called “Step 6: Set up video” (p. 116)
   - the section called “Step 7: Set up audio” (p. 117)
   - the section called “Step 8: Set up captions” (p. 118)

**Other fields**

The following field relates to implementing resiliency in an RTMP output:

- **RTMP settings** – **Input loss action** – For details about a field, 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. 224).

**Creating a UDP output group**

Follow these steps if, when you were planning the channel (p. 57), you determined that you want to include a UDP output group.

**Topics**

- Procedure to create a UDP output group (p. 115)
- Fields for the output destination (p. 115)
- Fields for the UDP transport (p. 116)
- Fields for the video, audio, and captions stream (encode) (p. 116)
- Fields for other UDP features (p. 116)
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. 115).
   - **UDP settings** – This section contains fields for setting up ID3 (p. 116) and for resiliency (p. 116).
   - **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. 115) and the connection to the destination (p. 115).
   - **Stream settings** – This section contains fields for the output streams (p. 116) (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. 118).

Fields for the output destination

The following fields configure the destination of the output:

- **Output – UDP destination** sections
- **Output – Output settings – Network settings – Buffer msec**

To specify the destination for the output

1. For the URLs for the destinations (A and B), specify two destinations when the channel is set up as a standard channel (p. 72), or one destination when it is set up as a single-pipeline channel. The URLs must use the RTP or UDP protocol and must include a port number.

   For example:
   
   \[\text{rtp://203.0.113.28:5000}\]

2. If you enable FEC (p. 116), leave space between the port numbers for the two destinations.

   For example, if one destination is \[\text{rtp://203.0.113.28:5000}\], assume that FEC also uses port 5002 and 5004. So the lowest possible port number for the other destination is 5005: \[\text{rtp://203.0.113.33:5005}\].

3. Complete Buffer msec 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 encodes in the output.

- **Stream settings** section

By default, the output is set up with one video encode and one audio encode. The output in a UDP group can contain one video encode, zero or more audio encodes, and zero or more captions encodes.

To set up the encodes

1. Decide if you need to create more encodes for this output, based on the workflow that you planned (p. 62).
2. Choose the appropriate **Add** button or **Delete** button to set up the output with the encodes—video, audio, and captions—that you planned for this output.
3. Complete the fields in each type of encode.

For information about the fields in each type of encode, see the following sections:

- the section called “Step 6: Set up video” (p. 116)
- the section called “Step 7: Set up audio” (p. 117)
- the section called “Step 8: Set up captions” (p. 118)

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, 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. 255).

Step 6: Set up the video encode

The output section for every type of output group contains a **Stream settings** section. In this section, you create **encodes** (p. 55) for the video, audio (if applicable), and captions (if applicable) in the output and specify the details of how you want these assets encoded.
The following procedure describes how to set up a video encode and assumes that you have created the output (p. 80) that will hold the video. The fields for a video encode are identical for all output group types. For example, the fields for video for an HLS group are identical to those for a UDP group.

**To set up the video encodes in most output types**

1. On the **Create channel** page, find the output group that you created (p. 80).
2. Under that output group, find the output (or the first 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 output. 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 **Color space** fields, see the section called “Color Space” (p. 246).
   - For information about the **Rate control** fields, see the section called “Video Rate Control Mode” (p. 323).
   - For information about the **Timecode** fields, see the section called “Timecode Configuration” (p. 321).
6. Continue setting up the audio encodes (p. 117), video encodes, and captions encodes (p. 118) for all outputs in all output groups. When you finish with the encodes for all outputs, go to save the channel (p. 118).

**To set up the video encodes in the Frame Capture output type**

1. On the **Create channel** page, find the output group that you created (p. 80).
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. 118).

### Step 7: Set up the audio encodes

The output section for every type of output group contains a **Stream settings** section. In this section, you create encodes (p. 55) for the video, audio, and captions in the output and specify the details of how you want these assets encoded.

The following procedure describes how to set up an audio encode and assumes that you have created the output (p. 80) that will hold the audio. The configuration options for an audio encode are identical for all output group types.

Note that a Frame Capture output group contains video but doesn't contain audio or captions.

**To set up the audio encodes in the outputs**

1. On the **Create channel** page, find the output group that you created (p. 80).
2. Under that output group, find the output (or the first output) where you want to set up the audio encode.
3. Choose the link for one of the audio encodes. (You might have created more than one encode.)
4. For **Codec settings**, choose the codec to use to encode this audio asset. The remaining fields change to match this codec.
5. Complete each field as appropriate. For details about a field, choose the **Info** link next to the field.
6. (Optional) Complete the fields in the **Remix settings** section, or keep the defaults (to omit remixing).
Step 8: Set up captions

7. (Optional) Complete the fields in the **Audio normalization** settings section, or keep the defaults (to omit normalization).
8. Repeat for each audio encode in this output, if any.
9. Continue setting up the audio encodes, **video encodes** (p. 116), and **captions encodes** (p. 118) for all outputs in all output groups. When you have finished with the encodes for all outputs, go to save the channel (p. 118).

**Step 8: Set up the captions encodes**

The output section for every type of output group contains a **Stream settings** section. In this section, you create encodes (p. 55) for the video, audio, and captions in the output and specify the details of how you want these assets encoded.

Note that a Frame Capture output group contains video but doesn't contain audio or captions.

Before setting up captions, see the section called “Reference: Supported Captions” (p. 325). There are rules about the following:

- What captions formats can be generated, depending on the format of the input captions and container. For example, ARIB can be generated only from ARIB input captions, which can appear only in a TS input.
- What captions type (embedded, object, sidecar) applies to which captions format. For example, DVB-Sub is an object type of format.
- Which format can be included in which type of output asset. For example, HLS can take WebVTT (a sidecar type), 608 (embedded), and burn-in.

**To set up the captions encodes in the outputs**

1. For detailed information about setting up captions, see the section called “Captions” (p. 224).
2. After you set up the captions for one output, continue setting up the **audio encodes** (p. 117), **video encodes** (p. 116), and captions encodes for all outputs in all output groups. When you have finished with the encodes for all outputs, go to save the channel (p. 118).

**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. 69).

- **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. 120).

- **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. 120).

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. 69).

**Topics**
- About Templates (p. 119)
- About Cloning (p. 120)
- Creating a Channel from a Template (p. 120)
- Creating a Channel by Cloning (p. 120)
- Creating a Custom Template (p. 121)

### 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. 121).

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
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. 119).
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. 119).
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. 120).)

To create a channel by cloning (console)

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)

2. In the navigation pane, choose Channels.
3. On the Channels page, choose the channel name (not the radio button).
4. On the details page, choose Create 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 edit a channel to change the input associated with that channel. Instead, you must clone (p. 120) the channel and associate a different input.
You can't edit a channel to change the channel class. Instead, you must update the channel class as its own operation. See the section called “Updating Channel Class” (p. 122), later in this chapter.

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. 69).
3. When done, choose Update channel.

Wait for the channel State to return to Idle before performing another action with this channel.
Editing the Tags Associated with a Channel

You can edit the tags associated with a channel at any time, when the channel is running or when it is idle. You can add more tags (up to the limit), and you can delete tags.

To edit the tags in a channel
1. On the Channels page, choose the channel name.
2. Choose the Tags tab. Add or delete tags. To edit the value of an existing tag, delete the tag and add it again. For more information, see the section called “Tagging Resources” (p. 319).
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 resiliency

You can change the channel class of an existing channel in order to enable or disable pipeline redundancy in the channel. For information the channel class, see the section called “Updating Channel Class” (p. 122).

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, Other channel actions, 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 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

You can change a single-pipeline channel to a standard channel, to implement pipeline redundancy. To change the channel class, the channel must be idle (not running).

There are two procedures for changing the channel class. Choose the procedure that matches how you planned (p. 245) the inputs and the channel.

- **Option A** – You set up a single-pipeline channel with standard inputs and upgrade potential. In this case, you completed both source URLs. Therefore when you upgrade the channel, you don’t need to edit the inputs to include a URL for the new pipeline.
- **Option B** – You set up a single-pipeline channel with single-pipeline inputs. In this case, you specified only one URL. Therefore when you upgrade, you must edit all of the inputs to include a URL for the second pipeline.

### Changing the class – option A

Follow this procedure if you set up a single-pipeline channel with standard inputs and upgrade potential. In this case, you completed both source URLs. Therefore when you upgrade the channel, you don’t need to edit the inputs to include a URL for the new pipeline.

**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. 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

Follow this procedure if you set up a single-pipeline channel with single-pipeline inputs. In this case, you specified only one URL. Therefore when you upgrade, you must edit all of the inputs to include a URL for the second pipeline.
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 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. 142) 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. 142) 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.

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)


2. In the navigation pane, choose Channels. (For information about the buttons on this page, see the section called “Editing a Channel” (p. 121), Starting, Stopping, and Pausing a Channel (p. 198), and the section called “Creating a Channel by Cloning” (p. 120).)

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 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. 57)—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 “Supported Input Types and Upstream Systems” (p. 336).

- **Live versus File** – An input is either a live (streaming) input or a file input. For more information, see the section called “Supported Input Types and Upstream Systems” (p. 336).

- **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 “Supported Input Types and Upstream Systems” (p. 336).

- **Input class** – An input is either a standard input or single-pipeline input:
  - You can use a standard input with a standard channel or a single-pipeline channel.
  - You can use a single-pipeline input only with a single-pipeline channel.
  
  For more information, see the section called “Channel class” (p. 243).

- **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. 262).

Inputs, input security groups, and channels

The input is one of the components of a workflow. The others are the input security group (p. 56) and the channel (p. 55). These three components are linked together. An input security group is attached to
Creating an input

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

Topics

- Getting ready (p. 127)
- Creating an HLS pull input (p. 127)
- Creating a MediaConnect push input (p. 129)
- Creating an MP4 pull input (p. 131)
- Creating an RTMP push input (p. 133)
- Creating an RTMP push input in Amazon VPC (p. 134)
- Creating an RTMP pull input (p. 137)
- Creating an RTP push input (p. 138)
- Creating an RTP push input in Amazon VPC (p. 139)

Getting ready

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

- **Setting up: Planning Your workflow (p. 57)** – You must coordinate your channel with your upstream system and downstream system, and you must design your outputs.
- the section called “Channel class” (p. 243) – You must decide if you want to implement pipeline redundancy—whether you set up a standard channel or a single-pipeline channel.
- the section called “Automatic input failover” (p. 218) – You must decide if you want to implement automatic input failover.

Creating an HLS pull input

Create your input before you create the channel that ingests the input.
To create an HLS pull input

1. From the upstream system, obtain the full URLs of the locations where MediaLive will pull the M3U8 manifest source from. You need two URLs for an input in a standard channel (p. 243), or one URL for an input in a single-pipeline channel.

The URLs will have one of these formats:

   • For a location that supports HTTP or HTTPS, enter an HTTP or HTTPS URL. For example, https://203.0.113.13/newschannel/anytownusa.m3u8 and https://203.0.113.54/newschannel/anytownusa.m3u8.

   • For a manifest that is stored on AWS Elemental MediaStore, the URL must include the data endpoint for the container. For example, the M3U8 file is called mlaw.m3u8, and it is stored in the container "movies" in the folder path premium/canada. The URL for the container might be eri39n.data.mediasstore.us-west-2.amazonaws.com. The value that you enter in this field would be mediastoressl://eri39n.data.mediasstore.us-west-2.amazonaws.com/premium/canada/mlaw.m3u8.

   • For a manifest that is stored on Amazon S3, enter the protocol as s3 or s3ssl, and then enter the bucket name and object for the manifest. For example, s3://movies/mlaw.m3u8 and s3://movies/redundant/mlaw.m3u8.

   With MediaLive, the Amazon S3 bucket name must not use dot notation. For example, mycompany-videos is acceptable but mycompany.videos isn't.


3. In the navigation pane, choose Inputs.

4. On the Inputs page, choose Create input.

5. In the Input details section, for Input name, enter a name.

6. For Input type, choose HLS.

7. In the Channel and input class section, choose the class for this input:

   • STANDARD

   • SINGLE-PIPELINE

   For information, see the section called "Channel class" (p. 243).

8. In the Input sources section, enter the URLs you previously obtained:

   • If the input is a standard input, complete both fields, to provide two URLs.

   • If the input is a single-pipeline input, complete the first field with the URL that you obtained and leave the second field empty.

9. 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. 31).

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. 319).

11. 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 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

There are three stages to creating a MediaConnect input in AWS Elemental MediaLive. First, you set up flows in AWS Elemental MediaConnect, then you create an input in MediaLive, then you verify that the output has been automatically created in AWS Elemental MediaConnect.

To create flows in AWS Elemental MediaConnect

1. Follow the procedure in Creating a Flow in the AWS Elemental MediaConnect User Guide to create one or two MediaConnect flows. You need two flows for an input in a standard channel (p. 243), or one flow for an input in a single-pipeline channel.

   Complete the fields as follows:

   - For Name, enter a name that includes A (for the first flow) and B (for the second flow). For example, sports_event_A and sports_event_B. Using these letters will help you match the flows to the input pipelines in AWS Elemental MediaConnect.

   - For Availability Zone, choose a different Availability Zone for each flow. (If you choose the same Availability Zone for both flows, MediaLive doesn't allow you to create the input.)

   - For the Source section, keep in mind that the source here is the source into AWS Elemental MediaConnect. It has nothing to do with MediaLive.

   - Don't create outputs or entitlements. In the next stage, MediaLive will automatically create outputs.

   - When you create the flow, AWS Elemental MediaConnect creates an ARN for each flow. For example:

     arn:aws:mediaconnect:us-west-1:111122223333:flow:1bgf67:sports_event_A and


   Note that the ARNs include the flow names as the last portion.

2. Look at the details of the flows and make sure that the two flows are in the same Region but different Availability Zones.

3. Make a note of the ARN or ARNs, and the Availability Zones. You need this information in order to set up the MediaLive input.

To create an input in 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 a name.

5. For Input type, choose MediaConnect.

6. In the Channel and input class section, choose the class for this input:

   - STANDARD
   - SINGLE-PIPELINE
For more information, see the section called “Channel class” (p. 243).

7. In the MediaConnect section, for ARN for flow A, specify the ARN for the flow that you identified as flow A.

   If you created a second flow, then for ARN for flow B, specify the ARN for flow B.

8. 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. 130).

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. 319).

10. 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.

11. 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 redundancy if one flow fails.

To verify the output in AWS Elemental MediaConnect

1. To view the list of outputs for the flow or flows that you created, follow the procedure in Viewing a List of Outputs of a Flow in the AWS Elemental MediaConnect User Guide.

   When you created the input in MediaLive, AWS Elemental MediaConnect automatically created outputs. Therefore, you end up with the following:

   Flow A (in MediaConnect) – output for flow A (in MediaConnect) – source for A (in MediaLive)

   Flow B (in MediaConnect) – output for flow B (in MediaConnect) – source for B (in MediaLive)

2. If you created two flows, make sure that there is one output for flow A and one output for flow B.

   Each output is assigned a name that consists of MediaLive dash random characters. For example:

   MediaLive-ace74fa23

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 “Supported Input Types and Upstream Systems” (p. 336).

**To create an MP4 pull input**

1. From the upstream system, obtain the full URLs of the locations where MediaLive will pull the MP4 file from. You need two URLs for an input in a standard channel (p. 243), or one URL for an input in a single-pipeline channel.
Creating an MP4 pull input

- For an upstream system that uses HTTP or HTTPS, enter an HTTP or HTTPS URL. For example:
  
  https://203.0.113.13/fillervideos/oceanwaves.mp4 and
  
  https://203.0.113.54/fillervideos/oceanwaves.mp4

- For a file that is stored on Amazon S3, enter the protocol as `s3` or `s3ssl`, and then enter the bucket name and object for the manifest. With MediaLive, the Amazon S3 bucket name must not use dot notation. For example, `mycompany-videos` is acceptable but `mycompany.videos` isn't.
  
  For example:
  
  `s3://fillervideos/main/oceanwaves.mp4` and `s3://fillervideos/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. 254). 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 later in this section.
  
  We recommend that you use the format `<protocol>$urlPath$`.


4. In the navigation pane, choose Inputs.

5. On the Inputs page, choose Create input.

6. In the Input details section, for Input name, enter a name.

7. For Input type, choose MP4.

8. In the Channel and input class section, choose the class for this input:

   - STANDARD
   - SINGLE-PIPELINE

   For more information, see the section called “Channel class” (p. 243).

9. In the Input sources section, enter the URLs (or the modified URLs) that you previously obtained:

   - If the input is a standard input, complete both fields, to provide two URLs.
   - If the input is a single-pipeline input, complete the first field with the URL that you obtained and leave the second field empty.

10. 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. 31).

11. 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. 319).

12. 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.

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
<th>Example</th>
<th>Example of the $urlPath$</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;protocol&gt;/$urlPath$</td>
<td>URL has only the protocol in the absolute portion</td>
<td>s3://$urlPath$</td>
<td>movies/my-movie.mp4</td>
</tr>
<tr>
<td>&lt;protocol and path&gt;/$urlPath$</td>
<td>URL has the protocol and path in the absolute portion</td>
<td>emsssl://movies/$urlPath$</td>
<td>my-movie.mp4</td>
</tr>
<tr>
<td>$urlPath$</td>
<td>URL has only the variable portion</td>
<td>$urlPath$</td>
<td>s3://movies/my-movie.mp4</td>
</tr>
</tbody>
</table>

**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 “Supported Input Types and Upstream Systems” (p. 336).

**To create an RTMP push input**

1. Obtain the ID of the input security group (p. 145) that you will use with this input. (Or if you will create an input security group at the same time as you create this input, obtain the IP addresses for the input security group.)
2. From the upstream system, obtain the application name and application instance (p. 60) that you identified when planning the workflow. You need two sources for an input in a standard channel (p. 243), or one source for an input in a single-pipeline channel.
4. In the navigation pane, choose Inputs.
5. On the Inputs page, choose Create input.
6. In the Input details section, for Input name, enter a name.
7. For Input type, choose RTMP (push).
8. In the Network mode section, choose Public.
9. 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. For more information about security groups, see MediaLive Input Security Groups (p. 145).
10. In the Channel and input class section, choose the class for this input:
    - STANDARD
    - SINGLE-PIPELINE
For more information, see the section called “Channel class” (p. 243).

11. 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 input, complete both fields, to specify two sources.
   - If the input is a single-pipeline input, complete the first field with the information that you obtained and leave the second field empty.

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. 319).

13. 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:


MediaLive always creates two endpoints:

   - If the channel for this input will be set up as a standard channel, both endpoints will be used.
   - If the channel for this input will be set up as a single-pipeline channel, only the first endpoint will be used. MediaLive won’t expect to receive content at the second endpoint.

14. Provide the upstream system with the following information:

   - If the channel for this input will be set up as a standard channel, provide both locations. The upstream system must push the video streams to these locations.
   - If the channel for this input will be set up as a single-pipeline channel, provide only the first location. The upstream system must push its one stream to this location.

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.

To set up the upstream system and the VPC

1. Identify a VPC. Make a note of the VPC ID. For example:

   vpc-3f139646

   Note
   We recommend that your organization create a VPC specifically for all AWS Media Services. A single VPC will help to ensure the availability of IP addresses, help in setting up appropriate rules in the security groups, help in adhering to the two Availability Zone rule, and help to ensure that a network administrator doesn’t accidentally delete elastic network interfaces.

2. Identify two subnets in the VPC. These rules apply to the subnets:

   - 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 MediaLive input.
• Any other RTP VPC inputs or RTMP VPC inputs that you create for use in the same channel as this input must be in subnets that are in the same Availability Zones as this input. The two subnets of the new inputs can be different from the existing input, but the Availability Zones of those two subnets must be the same as the Availability Zones of this input.

Make a note of the subnet IDs. For example:

subnet-1122aabb

subnet-4455ccdd

3. On your upstream system, set up two identical source streams. You need two source streams for an input in a standard channel (p. 243), or one source stream for an input in a single-pipeline channel.

If you have two source streams, set up one source stream so that it has an output interface in one of the subnets, and the other so that it has an output interface in the other subnet.

4. Make sure that the two streams are identical in terms of resolution and bitrate.

5. Identify at least one security group for each subnet.

   **Note**
   
   Don’t confuse the security groups that belong to Amazon VPC with the input security groups that belong to MediaLive.

   These rules apply to the security groups for each subnet:

   • 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 3500.

6. Make a note of the IDs of the security group or groups. For example:

    sg-51530134

7. Make a note of the following three characteristics of the source stream. You will need this information to set up the channel:

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

**To create an RTMP VPC push input**


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 a name.

5. For **Input type**, choose **RTMP (push)**.

6. In the **Network mode** section, choose **VPC**.

7. In the **VPC settings** section, choose **Use existing VPC**. For **Subnets**, choose one of the subnets that you identified. 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`.)

8. Choose the second subnet. This second time, the dropdown list shows only the subnets in the same VPC as the first subnet.

9. For **Security groups**, choose the security group or groups that you identified, 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`

10. 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. 141).

11. In the **Channel and input class** section, choose the class for this input:

   - STANDARD
   - SINGLE-PIPELINE

   For more information, see the section called “Channel class” (p. 243).

12. 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 input, complete both fields, to specify two sources.
   - If the input is a single-pipeline input, complete the first field with the information that you obtained and leave the second field empty.

13. 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. 319).

14. 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:

   ```rtmp://10.99.20.15:1935```

   ```rtmp://192.0.2.53:1935```

15. Provide the upstream system with these endpoints:

   - If the channel for this input will be set up as a standard channel, provide both endpoints. The upstream system must push the content to both endpoints.
   - If the channel for this input will be set up 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 RTMP input has two IP addresses. Each address is in one of those same subnets. In this way, the delivery of the content from the upstream system to MediaLive takes place within the security of the VPC.

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. In order 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 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 “Supported Input Types and Upstream Systems” (p. 336).

To create an RTMP pull input

1. From the upstream system, obtain the full URLs of the locations where MediaLive will pull the source from. You need two URLs for an input in a standard channel (p. 243), or one URL for an input in a single-pipeline channel. For example:

   rtmp://203.0.113.20:1935/movies/classic


3. In the navigation pane, choose Inputs.

4. On the Inputs page, choose Create input.

5. In the Input details section, for Input name, enter a name.

6. For Input type, choose RTMP (pull).

7. In the Channel and input class section, choose the class for this input:

   • STANDARD
   • SINGLE-PIPELINE

   For more information, see the section called “Channel class” (p. 243).

8. In the Input sources section, enter the URLs that you previously obtained:

   • If the input is a standard input, complete both fields, to provide two URLs.
   • If the input is a single-pipeline input, complete the first field with the URL that you obtained and leave the second field empty.

9. Enter the user name and password key for accessing the RTMP 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. 31).

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. 319).

11. 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 the channel for this input will be 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.
   • If the channel for this input will be 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 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 “Supported Input Types and Upstream Systems” (p. 336).
Creating an RTP push input

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 a name.
5. For Input type, choose RTP.
6. In the Network mode section, choose Public.
7. 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 MediaLive Input Security Groups (p. 145).
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. 319).
9. Choose Create.

MediaLive creates the input and automatically creates two endpoints on that input. These endpoints include the port 5000. For example:

```
rtp://203.0.113.19:5000
rtp://203.0.113.131:5000.
```

MediaLive always creates two endpoints:

- If the channel for this input will be set up as a standard channel, both endpoints will be used.
- If the channel for this input will be set up as a single-pipeline channel, only the first endpoint will be used. MediaLive won't expect to receive content at the second endpoint.

10. Provide the upstream system with the following information:

- If the channel for this input will be set up as a standard channel, provide both locations. The upstream system must push the video streams to these locations.
- If the channel for this input will be set up as a single-pipeline channel, provide only the first location. The upstream system must push its one stream to this location.

Creating an RTP push input in Amazon VPC

You create an RTP push input in Amazon Virtual Private Cloud 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 “Supported Input Types and Upstream Systems” (p. 336).

To set up the upstream system and the VPC

1. Identify a VPC. Make a note of the VPC ID. For example:

```
vpc-3f139646
```

**Note**

We recommend that your organization create a VPC specifically for all AWS Media Services. A single VPC will help to ensure the availability of IP addresses, help in setting up appropriate rules in the security groups, help in adhering to the two Availability Zone rule, and help to ensure that a network administrator doesn't accidentally delete elastic network interfaces.
2. Identify two subnets in the VPC. These rules apply to the subnets:
   - 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 MediaLive input.
   - Any other RTP VPC inputs or RTMP VPC inputs that you create for use in the same channel as this input must be in subnets that are in the same Availability Zones as this input. The two subnets of the new inputs can be different from the existing input, but the Availability Zones of those two subnets must be the same as the Availability Zones of this input.

   Make a note of the subnet IDs. For example:

   subnet-1122aabb
   subnet-4455ccdd

3. On your upstream system, set up two identical source streams (if the channel for this input will be set up as a standard channel (p. 243)) or one source stream (if it will be set up as a single-pipeline channel). Set up one source stream so that it has an output interface in one of the subnets, and the other so that it has an output interface in the other subnet.

4. Make sure that the two streams are identical in terms of resolution and bitrate.

5. Identify at least one security group for each subnet.

   Note
   Don’t confuse the security groups that belong to Amazon VPC with the input security groups that belong to MediaLive.

   These rules apply to the security groups for each subnet:
   - 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.

6. Make a note of the IDs of the security group or groups. For example:

   sg-51530134

7. Make a note of the following three characteristics of the source stream. You will need this information to set up the channel:
   - The video codec
   - The resolution of the video: SD, HD, or UHD.
   - The maximum input bitrate

To create an RTP VPC push input from Amazon VPC

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 a name.
5. For Input type, choose RTP.
6. In the Network mode section, choose VPC.
7. In the VPC settings section, choose Use existing VPC. For Subnets, choose one of the subnets that you identified. The dropdown list shows subnets in all VPCs, identified as follows:
Creating an RTP push input in Amazon VPC

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.)

8. Choose the second subnet. This second time, the dropdown list shows only the subnets in the same VPC as the first subnet.

9. For Security groups, choose the security group or groups that you identified, 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
```

10. 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. 141).

11. 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. 319).

12. 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://192.0.2.44:5000.
```

13. Provide the upstream system with these endpoints:

- If the channel for this input will be set up as a standard channel, provide both endpoints. The upstream system must push the content to both endpoints.
- If the channel for this input will be set up 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. Each address is in one of those same subnets. In this way, the delivery of the content from the upstream system to MediaLive takes place within the security of the VPC.

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. In order 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:

- You can attach a different input security group.
- 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 input endpoint. To change these addresses, you must delete the input and create it again.
For a MediaConnect push input, you can edit the ARNs to refer to different MediaConnect flows. The outputs for the former ARNs will be deleted in AWS Elemental 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.

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.

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

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 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. 319).
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

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. 122).

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 is a set of one or more whitelist rules. Each rule is a range of IP addresses. These IP addresses are allowed to push traffic to the input destinations of a channel (to push traffic to the channel's input).

Topics

• Purpose of an Input Security Group (p. 145)
• Creating an Input Security Group (p. 145)
• Editing an Input Security Group (p. 146)
• Deleting an Input Security Group (p. 146)

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, or MediaConnect inputs. These inputs implement security in other ways.

For the relevant push inputs, an input security group restricts access to the input and prevents unauthorized third parties from pushing content into a channel that is associated with that input. Without the protection of this feature, any third party could push content to an MediaLive input if they know the 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; one input security group can "serve" several inputs.

Creating an Input Security Group

You create an input security group and then attach it to a "push" input when you create or edit that input.

To create an input security group

2. In the navigation pane, choose Input security groups.
3. On the Input security groups page, choose Create input security group.
4. For New security group, type one or more IPv4 CIDR blocks. Each CIDR block must include a subnet mask. Separate the entries with commas, or type each entry on a separate line.

Each item in the list represents one whitelist rule, even if it encompasses several individual addresses. For example, each of the following examples counts as one rule:

192.0.2.0/24
192.0.2.111/32
5. 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. 319).
6. Choose Create.

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

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

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. 319).

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

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.
MediaLive Multiplex

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. 290).

Topics

- Summary of Actions (p. 148)
- Creating a Multiplex and Program (p. 149)
- Creating a Channel (p. 150)
- Editing Multiplexes, Programs, and Channels (p. 150)
- Deleting Multiplexes, Programs, and Channels (p. 151)

Summary of Actions

The following table summarizes the create, edit, and delete capabilities for the multiplex, program, and channel.

<table>
<thead>
<tr>
<th>Item</th>
<th>Action</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiplex</td>
<td>Create</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Edit</td>
<td>The multiplex must be idle, and must not have any associated programs.</td>
</tr>
<tr>
<td>Program</td>
<td>Create</td>
<td>The multiplex for the program can be idle or running.</td>
</tr>
<tr>
<td></td>
<td>Edit</td>
<td>The multiplex for this program can be idle or running. The channel for this program can be idle or running.</td>
</tr>
</tbody>
</table>
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
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
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
2. In the navigation pane, choose Multiplexes, and then choose the multiplex where you want to add the program.
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. 293).

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

Editing a Program

You can edit a program at any time, including when the multiplex is running or when the associated channel is running.

To edit a program
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. 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. 297).
2. Edit the channel. For more information, see the section called “Editing a Channel” (p. 121).

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
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.
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

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. 122).
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 (p. 153)
- Add-on Reservations (p. 155)
- Purchasing a Reservation (p. 157)
- Viewing Purchased Reservations (p. 158)
- Deleting a Reservation (p. 158)

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.

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 bit rate of 29.97 fps in the channel configuration falls within the range of a bit rate attribute of \( \leq 30 \text{ fps} \) 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 "\( \leq 30 \text{fps} \)".
• 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".

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

How an Input or Output Reservation Is Applied

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

At the end of the cycle, AWS applies the minutes from a given reservation to reduce the cost for the processed items (inputs or outputs) whose attributes match this reservation. For each minute in the month, AWS determines if one or more matching items 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. 154), 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. 154), 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)**

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. 157).
4. Choose an offering, and then 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.)
5. To view the cart contents, choose the **Cart** tab.
6. 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)**

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)**

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 inserting actions into the schedule that is associated with the channel.

**Supported Actions**

You can use actions to do the following:

- Switch the input that the running channel is ingesting.
- Insert a static image overlay (an image layered over the underlying video) 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 “Types of Actions” (p. 160).

**Supported Action Timings**

You can set up an action to occur at a specific (fixed) time, as soon as possible (immediate), or to follow when the currently running input is finished.

For information on the timings applicable to each action, see the section called “Types of Timing” (p. 161).

**Required Reading**

This chapter describes how to set up the schedule with the actions that you want. Before you get started, we recommend that you do the following:

- If you want to switch inputs, read the information about setting up a channel to include multiple inputs, as described in the section called “Input Switching” (p. 262).
- If you want to activate and deactivate image overlays, set up the images that you will use. See the section called “Image Overlays” (p. 260).
- If you want to insert SCTE-35 messages, read the information that explains how MediaLive handles SCTE-35 messages in transport stream (TS) outputs. See the section called “SCTE-35 Message Processing” (p. 299).
- If you want to insert ID3 metadata, read the information about the options for ID3 metadata in MediaLive. See the section called “ID3 Metadata” (p. 255).
- If you want to insert ID3 segment tags, enable the feature. See the section called “ID3 Segment Tags” (p. 258).
Types of Actions in the Schedule

The schedule is a list of actions that a channel executes as it is running. You create the actions in the schedule, and you set either a fixed or relative time that specifies when you want the channel to execute each action.

MediaLive stores the actions in the schedule. Shortly before its start time, each action is received by the channel. At the action start time, the channel executes the action.

You can delete an action from the schedule up to the moment that the channel receives it.

Types of Schedule Actions for Input Switching

• Action to switch the input that the running channel is ingesting.

  The action specifies when to execute the switch—either at a specific (fixed) time, as soon as possible (immediate), or to follow when the currently running input is finished.

  When the channel executes this action, the channel stops ingesting the current input and starts ingesting the specified input.

  With input switching, the channel must already be set up with all the inputs that you want to switch to. You can’t use a switch action to insert an input into the list of inputs in the channel. You can use it only to switch among the inputs that are already in the list.

  After you create an action, the action waits in the schedule. The channel receives the action 15 seconds before the action's designated start time. At the start time, the channel executes the action.

Before you add input switching actions to the schedule, read the section called “Input Switching” (p. 262).

Types of Schedule Actions for Image Overlay

• Action to activate a static image overlay. An image overlay is an image that is layered over the underlying video.

  When a channel executes this action, the image overlay is superimposed on the video. If the image overlay information includes a duration, then at the appropriate time the image overlay is removed.

• Action to deactivate a static image overlay.

  When the channel executes the deactivate action, the image overlay is removed. You therefore use this action to remove a currently running image overlay before the specified duration, or remove it when no duration is specified.

  Note
  You remove the active image from the video by creating a deactivate action. You don’t deactivate the image by deleting the activate action from the schedule. (In fact, deleting the action has no effect because its start time has passed.)

Before you add image overlay actions to the schedule, read the section called “Image Overlays” (p. 260).
Types of Schedule Actions for SCTE-35

- 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.

MediaLive processes the inserted message in the same way as it processes messages that were already in the source content.

Before you add SCTE-35 actions to the schedule, read the section called “SCTE-35 Message Processing” (p. 299).

Types of Schedule Actions for ID3 Metadata and Tags

- Action to insert ID3 metadata in outputs where ID3 passthrough is enabled (p. 255).
  
  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. 255).
  
  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. 255).
Before you add ID3 segment tag actions to the schedule, read the section called “ID3 Segment Tags” (p. 258).

Types of Schedule Actions for Pause and Unpause

- Action to pause the specified pipeline or pipelines in the channel.

  By implication, any pipelines that you don't include in the action are automatically set to unpause. You must specify a start time for the pause action.

- Action to unpause a pipeline or pipelines that are currently paused.

  You must specify a start time for the unpause action.

  **Note**

  You unpause (stop the pause) on the pipeline by creating an unpause action. You don't stop the pause by deleting the pause action from the schedule. (In fact, deleting the action has no effect because its start time has passed.)

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.
The time that you specify must be at least 15 seconds in the future from the time that you create the action.

- Immediate – Perform the action as soon as possible.
  
  You don't specify a time.

- Follow – Perform the action when the currently running input is finished.
  
  The currently running input 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.

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.

<table>
<thead>
<tr>
<th>Type of action</th>
<th>Supported types of timing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch the input (perform an input switch)</td>
<td>Yes</td>
</tr>
<tr>
<td>Activate a static image overlay</td>
<td>Yes</td>
</tr>
<tr>
<td>Deactivate a static image overlay</td>
<td>Yes</td>
</tr>
<tr>
<td>Insert a SCTE-35 message</td>
<td>Yes</td>
</tr>
<tr>
<td>Insert ID3 metadata</td>
<td>Yes</td>
</tr>
<tr>
<td>Insert an ID3 segment tag</td>
<td>Yes</td>
</tr>
<tr>
<td>Pause or unpause one or both pipelines</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Note A**

With a follow, the "Yes" means that this type of action can follow an input switch. Therefore, the action that is being followed is always an input. The action that does the follow is an input switch or a SCTE-35 message.

## Working with the Schedule (Console)

You can use the AWS Elemental MediaLive console to create or delete any of the schedule actions (p. 160) in a channel. You can work with the schedule when the channel is running or when it is idle.

The actions are executed in the channel when the channel is running. The channel executes the actions at the start times that you specify.

The console provides two views for working with actions:

- A list view that lists actions in tabular format
Creating Actions

- 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) (p. 163)
- Deleting Actions from the Schedule (Console) (p. 173)
- Modifying Actions in the Schedule (Console) (p. 174)
- Viewing the Schedule (Console) (p. 176)

Creating Actions in the Schedule (Console)

You can create actions to switch the input that the channel is ingesting, activate or deactivate image overlays on the video, insert SCTE-35 messages in the output, insert ID3 metadata or segment tags in the output, or pause or unpause a pipeline in the channel.

You can create an action in the schedule when the channel is running or when it is idle.

The action that you create must have a UTC start time or a follow start time (for input switches) that is at least 15 seconds in the future. (In other words, it must not be already received in the channel.) After that cutoff, AWS Elemental MediaLive rejects the create request.

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

To create an action

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. 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. 176).
5. 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.
6. 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.
7. On the Create schedule action page, complete the fields. For information about completing the fields, see the following topics.

8. 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 “Types of Switches” (p. 265).

Topics
• Fields for an Input Switch (p. 164)
• Fields for Activating an Image Overlay (p. 168)
• Fields for Deactivating an Image Overlay (p. 168)
• Fields for a Splice_Insert Message (p. 169)
• Fields for a Time_Signal Message (p. 170)
• Fields for a Return-to-Network Message (p. 171)
• Fields for ID3 Metadata (p. 172)
• Fields for ID3 Segment Tags (p. 172)
• Fields for Pause (p. 173)
• Fields for Unpause (p. 173)

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.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action name</td>
<td>A name for this input switch.</td>
</tr>
<tr>
<td>Action type</td>
<td>Input Switch.</td>
</tr>
<tr>
<td>Input attachment</td>
<td>The input to switch to. The input must already be set up as an input attachment (p. 73) in this channel. It can be a live input or a file input. For example, if you want to switch from input A to input B, specify input B in this field.</td>
</tr>
<tr>
<td>Start type</td>
<td>Fixed, Immediate, or Follow.</td>
</tr>
<tr>
<td>Date and time</td>
<td>If the Start type is Fixed, specify the date and time (in UTC format) that the channel must switch</td>
</tr>
</tbody>
</table>
### Creating Actions

#### 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 “Types of Switches” (p. 265).

For example, if you want to switch from input A to input B, specify input A in this field.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action name</td>
<td>A name for this input switch.</td>
</tr>
<tr>
<td>Action type</td>
<td>Input Switch.</td>
</tr>
<tr>
<td>Input attachment</td>
<td>The input to switch to. The input must already be set up as an input attachment (p. 73) in this channel. It can be a live input or a file input. For example, if you want to switch from input A to input B, specify input B in this field.</td>
</tr>
<tr>
<td>Input clippings settings – Enable input clipping</td>
<td>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.</td>
</tr>
<tr>
<td>Input clippings settings – Input timecode source</td>
<td>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</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action name</td>
<td>A name for this input switch.</td>
</tr>
<tr>
<td>Action type</td>
<td>Input Switch.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Input attachment</strong></td>
<td>The input to switch to. The input must already be set up as an input attachment (p. 73) in this channel. It can be a live input or a file input.</td>
</tr>
<tr>
<td></td>
<td>For example, if you want to switch from input A to input B, specify input B in this field.</td>
</tr>
<tr>
<td><strong>Dynamic input setting – URL path for input source A</strong></td>
<td>This field appears if the input is a dynamic input. 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.</td>
</tr>
<tr>
<td><strong>Dynamic input setting – Use the same URL path for input source B</strong></td>
<td>This field appears if the input is attached to a standard channel, meaning that it has two pipelines and therefore has two sources.</td>
</tr>
<tr>
<td></td>
<td>Enable this field (the default) if you want to use the same value for the $urlPath$ in source A and source B.</td>
</tr>
<tr>
<td></td>
<td>Disable this field to use a different value, then enter the value.</td>
</tr>
<tr>
<td><strong>Input clippings settings – Enable input clipping</strong></td>
<td>This field appears only for a file input.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Input clippings settings – Input timecode source</strong></td>
<td>Choose the source:</td>
</tr>
<tr>
<td></td>
<td>Zero-based – To set the start and end times relative to the start of the file, which is 00:00:00:00.</td>
</tr>
<tr>
<td></td>
<td>Embedded – To set the times based on the timecode in the file. The file must have a timecode, otherwise the clipping instruction is ignored.</td>
</tr>
<tr>
<td><strong>Input clippings settings – Start timecode, Stop timecode</strong></td>
<td>Complete one or both fields. Enter values in the format hh:mm:ss:ff.</td>
</tr>
<tr>
<td><strong>Input clippings settings – Last frame clipping behavior</strong></td>
<td>This field appears only if you specify a stop timecode.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Include last frame – Don't clip the file. In the preceding example, include frame 19.</td>
</tr>
<tr>
<td><strong>Start type</strong></td>
<td>Fixed, Immediate, or Follow.</td>
</tr>
</tbody>
</table>
Fields for Activating an Image Overlay

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

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action type</td>
<td>Static Image Activate.</td>
</tr>
<tr>
<td>Action name</td>
<td>A name for this activation action. For example, the name of the image to overlay.</td>
</tr>
<tr>
<td>Start type</td>
<td>Fixed.</td>
</tr>
<tr>
<td>Date and time</td>
<td>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.</td>
</tr>
<tr>
<td>Other fields</td>
<td>Complete these fields to control the behavior of the activation.</td>
</tr>
</tbody>
</table>

Fields for Deactivating an Image Overlay

This table shows the fields that apply for an action to deactivate an image overlay.
## Creating Actions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action type</td>
<td>Static Image Deactivate.</td>
</tr>
<tr>
<td>Action name</td>
<td>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 &quot;deactivate.&quot;</td>
</tr>
<tr>
<td>Start type</td>
<td>Fixed.</td>
</tr>
<tr>
<td>Date and time</td>
<td>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.</td>
</tr>
</tbody>
</table>

### Fields for a Splice_Insert Message

This table shows the fields that apply for an action to insert a splice_insert SCTE-35 message.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action type</td>
<td>SCTE-35 Splice Insert.</td>
</tr>
<tr>
<td>Action name</td>
<td>A name for this splice_insert action. For example, splice_insert actions could be numbered sequentially, restarting every day or every month.</td>
</tr>
<tr>
<td>Start type</td>
<td>Fixed or Follow or Immediate.</td>
</tr>
<tr>
<td>Date and time</td>
<td>If the Start type is Fixed, specify the UTC start time for the splice_insert action. The time should be at least 15 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.</td>
</tr>
<tr>
<td>Reference action name</td>
<td>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 “Types of Switches” (p. 265).</td>
</tr>
<tr>
<td>Follow point</td>
<td>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.</td>
</tr>
</tbody>
</table>
| Splice event id     | The ID for the splice event. Enter an ID for the splice event that is unique among all scheduled
Creating Actions

Field Description

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.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action type</td>
<td>SCTE-35 Time Signal.</td>
</tr>
<tr>
<td>Action name</td>
<td>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.</td>
</tr>
<tr>
<td>Start type</td>
<td>Fixed or Follow or Immediate.</td>
</tr>
<tr>
<td>Date and time</td>
<td>If the Start type is Fixed, specify the UTC start time for the time_signal. The time should be at least 15 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.</td>
</tr>
<tr>
<td>Reference action name</td>
<td>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</td>
</tr>
</tbody>
</table>
### Creating Actions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Field                        | **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 “Types of Switches” (p. 265). |
| **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.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Action type</strong></td>
<td><strong>SCTE-35 Return to Network.</strong></td>
</tr>
<tr>
<td><strong>Action name</strong></td>
<td>A name for this return-to-network action. For example, <em>splice0003_ return_early</em>.</td>
</tr>
<tr>
<td><strong>Start type</strong></td>
<td><strong>Fixed</strong> or <strong>Follow</strong> or <strong>Immediate</strong>.</td>
</tr>
</tbody>
</table>
| **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 later than the time that you submit the action.  
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 “Types of Switches” (p. 265). |
| **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.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action type</td>
<td>HLS Timed Metadata.</td>
</tr>
<tr>
<td>Action name</td>
<td>A name for the metadata item. You might want to design a convention for naming ID3 metadata items, such as id3_metadata-&lt;UTC time&gt;.</td>
</tr>
<tr>
<td>Start type</td>
<td>Fixed.</td>
</tr>
<tr>
<td>Date and time</td>
<td>The UTC start time for the ID3 metadata item. The time should be at least 15 seconds later than the time that you submit the action.</td>
</tr>
<tr>
<td></td>
<td>Note that the time is the wall clock time, not the timecode in the input.</td>
</tr>
<tr>
<td>Id3</td>
<td>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.</td>
</tr>
</tbody>
</table>

Fields for ID3 Segment Tags

This table shows the fields that apply for an action to insert ID3 segment tags.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action type</td>
<td>HLS ID3 Segment Tagging.</td>
</tr>
<tr>
<td>Action name</td>
<td>A name for the segment tag.</td>
</tr>
<tr>
<td>Start type</td>
<td>Fixed.</td>
</tr>
<tr>
<td>Date and time</td>
<td>The UTC start time for the ID3 segment tag. The time should be at least 15 seconds later than the time that you submit the action.</td>
</tr>
<tr>
<td></td>
<td>Note that the time is the wall clock time, not the timecode in the input.</td>
</tr>
<tr>
<td>Tag</td>
<td>The content of the tag. The content is plaintext.</td>
</tr>
<tr>
<td></td>
<td>The content can include MediaLive variable data (p. 343). 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.</td>
</tr>
<tr>
<td></td>
<td>$dt$#$sn$</td>
</tr>
</tbody>
</table>
# Fields for Pause

In **Schedule action settings**, complete the following fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Action type</strong></td>
<td>Pause</td>
</tr>
<tr>
<td><strong>Action name</strong></td>
<td>A name for the action</td>
</tr>
<tr>
<td><strong>Start type</strong></td>
<td>Fixed</td>
</tr>
<tr>
<td><strong>Date and time</strong></td>
<td>The UTC start time for the action. The time should be at least 15 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.</td>
</tr>
<tr>
<td><strong>Actions</strong></td>
<td>Choose <a href="#">Add actions</a>, then for <strong>Pipeline id</strong>, choose <strong>PIPELINE_0</strong> or <strong>PIPELINE_1</strong>.</td>
</tr>
</tbody>
</table>

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 executed.

# Fields for Unpause

In **Schedule action settings**, complete the following fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Action type</strong></td>
<td>Pause</td>
</tr>
<tr>
<td><strong>Action name</strong></td>
<td>A name for the action</td>
</tr>
<tr>
<td><strong>Start type</strong></td>
<td>Fixed</td>
</tr>
<tr>
<td><strong>Date and time</strong></td>
<td>The UTC start time for the action. The time should be at least 15 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.</td>
</tr>
<tr>
<td><strong>Actions</strong></td>
<td>Keep this section empty. Don't add any actions.</td>
</tr>
</tbody>
</table>

When you choose **Create**, MediaLive adds an action to the schedule to unpause all pipelines.

## Deleting Actions from the Schedule (Console)

You can delete actions that you previously created in the schedule, so long as the action has not been received by the channel. You can delete an action when the channel is running or when it is idle.
Note
If the channel has already received the action, you might be able to modify it to effectively delete it. For more information, see Modifying or Deleting an ID3 Segment Tag That Is in Progress (p. 176).

The action that you delete must have a UTC start time or a follow start time (for input switches) that is at least 15 seconds in the future. If the channel has already received the action, AWS Elemental MediaLive rejects the delete request.

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

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. 176).
5. Choose one or more actions to delete. Make sure that you choose actions with a UTC start time in the future.

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)

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. 176).
5. In each action section, choose the X to delete the action. Make sure that you choose actions with a UTC start time in the future.

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
- Modifying Actions Not Yet Started (p. 175)
- Modifying an Input Switch Action (p. 175)
- Modifying an Input Switch Action in a Follow Chain (p. 175)
• Inserting an Input Switch Action into a Follow Chain (p. 176)
• Modifying an Image Overlay That Is in Progress (p. 176)
• Modifying a SCTE-35 Message That Is in Progress (p. 176)
• Modifying or Deleting an ID3 Segment Tag That Is in Progress (p. 176)

Modifying Actions Not Yet Started

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 these deadlines.

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:

<table>
<thead>
<tr>
<th>Input</th>
<th>Type</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Fixed</td>
<td>File</td>
</tr>
<tr>
<td>B</td>
<td>Follow</td>
<td>File</td>
</tr>
<tr>
<td>C</td>
<td>Follow</td>
<td>File</td>
</tr>
<tr>
<td>D</td>
<td>Follow</td>
<td>File or Live</td>
</tr>
<tr>
<td>E</td>
<td>Fixed</td>
<td>File or Live</td>
</tr>
</tbody>
</table>

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) 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:

<table>
<thead>
<tr>
<th>Input</th>
<th>Type</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Fixed</td>
<td>File</td>
</tr>
<tr>
<td>B</td>
<td>Follow</td>
<td>File</td>
</tr>
<tr>
<td>C</td>
<td>Follow</td>
<td>File</td>
</tr>
<tr>
<td>D</td>
<td>Follow</td>
<td>File or Live</td>
</tr>
<tr>
<td>E</td>
<td>Fixed</td>
<td>File or Live</td>
</tr>
</tbody>
</table>
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 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 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.

**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)

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 input switch. The remaining input switches in that card are always follow 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 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 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 possible to query MediaLive for the duration.
- Actions that relate to input switching: There is no impact.
- 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.
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 (p. 178)
- Batch Command in Different Interfaces (p. 179)
- JSON Payload in Different Interfaces (p. 179)

**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 a UTC 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 execute 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
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 \( \text{CREATES} \) section and a \( \text{DELETES} \) 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. 195) 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 rules apply to create actions:

- You can create actions when the channel is running or when it is idle.
- The action must have a UTC start time that is at least 15 seconds in the future (and no more than 14 days in the future). After that cutoff, AWS Elemental MediaLive rejects the create request.
- 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.
- The action that you delete must have a UTC start time or a follow start time (for input switches) that is at least 15 seconds in the future. (In other words, it must not be already received in the channel.) After that cutoff, AWS Elemental MediaLive rejects the delete request.
- 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. 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.
2. 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.
3. Save the file to the folder where you are running the AWS CLI.
4. 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
   ```

5. To submit the command, press Enter. The response appears on the screen. The response repeats the data from the request.

   If the JSON payload includes create actions and you didn't enter `ActionName` elements for an action, AWS Elemental MediaLive generates one and includes it in the response.

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 (p. 181)
- Activate Image Action – Payload (p. 184)
- Deactivate Overlay Action – Payload (p. 185)
- Splice_Insert Message – Payload (p. 186)
- Time_Signal Message – Payload (p. 187)
- Return-to-Network Message – Payload (p. 189)
- ID3 Metadata Item – Payload Item (p. 190)
- ID3 Segment Tag – Payload Item (p. 191)
- Pause Pipeline Action – Payload (p. 192)
- Combination of Create Actions (p. 193)

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"
        }
      }
    },
    {"ImmediateModeScheduleActionStartSettings": {}}
    {"FollowModeScheduleActionStartSettings": {
      "FollowPoint": "string",
      "ReferenceActionName": "string"
    }}
  },
  "ActionName": "string",
  "ScheduleActionSettings": {
    "InputSwitchSettings": {
      "InputAttachmentNameReference": "string",
      "UrlPath": [
        "string",
        "string"
      ],
      "InputClippingSettings": {
        "InputTimecodeSource": "string",
        "StartTimecode": {
          "Timecode": "string"
        },
        "StopTimecode": {
          "Timecode": "string",
          "LastFrameClippingBehavior": "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": "string"
        }
      }
    }
  }
}
```
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:

```json
{
  "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`.

```json
{
  "ChannelId": "999999",
  "Creates": {
    "ScheduleActions": [
      {
      }
    ]
  }
}
```
"ScheduleActionStartSettings": {
  "ImmediateModeScheduleActionStartSettings": {
  },
  "ActionName": "action_unscheduled_slate",
  "ScheduleActionSettings": {
    "InputSwitchSettings": {
      "InputAttachmentNameReference": "slate",
      "UrlPath": [
        "main/oceanwaves.mp4",
        "redundant/oceanwaves.mp4"
      ]
    }
  }
},
"ActionName": "string",
"ScheduleActionSettings": {
  "StaticImageActivateSettings": {
    "Duration": "integer",
    "Image": {
      "PasswordParam": "string",
      "Uri": "string",
      "Username": "string"
    },
    "FadeOut": "integer",
    "ImageY": "integer",
    "FadeIn": "integer",
    "ImageX": "integer",
    "Width": "integer",
    "Opacity": "integer",
    "Layer": "integer",
    "Height": "integer"
  }
}

**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:

```json
{
  "ChannelId": "string",
  "Creates": {
    "ScheduleActions": [
      {
        "ScheduleActionStartSettings": {
          "FixedModeScheduleActionStartSettings": {
            "Time": "string"
          }
        },
        "ActionName": "string",
        "ScheduleActionSettings": {
          "StaticImageActivateSettings": {
            "Duration": "integer",
            "Image": {
              "PasswordParam": "string",
              "Uri": "string",
              "Username": "string"
            },
            "FadeOut": "integer",
            "ImageY": "integer",
            "FadeIn": "integer",
            "ImageX": "integer",
            "Width": "integer",
            "Opacity": "integer",
            "Layer": "integer",
            "Height": "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://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"
          }
        },
        "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:

```json
{
  "ChannelId": "999999",
  "Creates": {
    "ScheduleActions": [
      {
        "ScheduleActionStartSettings": {
          "FixedModeScheduleActionStartSettings": {
            "Time": "2018-05-21T20:42:04.000Z"
          }
        },
        "ActionName": "stop_overlay_029",
        "ScheduleActionSettings": {
          "StaticImageDeactivateSettings": {
            "FadeOut": 500,
            "Layer": 1
          }
        }
      }
    ]
  }
}
```

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:

```json
{
  "ChannelId": "string",
  "Creates": {
    "ScheduleActions": [
      {
        "ScheduleActionStartSettings": {
          "FixedModeScheduleActionStartSettings": {
            "Time": "string"
          }
        },
        "ActionName": "string",
        "ScheduleActionSettings": {
          "Scte35SpliceInsertSettings": {
            "SpliceEventId": "integer",
            "Duration": "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).
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.

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.
"Creates": {
  "ScheduleActions": [
    { "ScheduleActionStartSettings": {
      "FixedModeScheduleActionStartSettings": {
        "Time": "2018-05-21T20:42:04.000Z"
      }
    },
    "ActionName": "adavail_3708",
    "ScheduleActionSettings": {
      "Scte35TimeSignalSettings": {
        "Scte35Descriptors": [ {
          "Scte35DescriptorSettings": {
            "SegmentationDescriptorScte35DescriptorSettings": {
              "SubSegmentsExpected": "integer",
              "SegmentationEventId": "integer",
              "SegmentationDuration": "integer",
              "SegmentationCancelIndicator": "enum",
              "SubSegmentNum": "integer",
              "SegmentationUpidType": "integer",
              "SegmentNum": "integer",
              "SegmentationCancelIndicator": "enum",
              "DeliveryRestrictions": {
                "DeviceRestrictions": "enum",
                "WebDeliveryAllowedFlag": "enum",
                "NoRegionalBlackoutFlag": "enum",
                "ArchiveAllowedFlag": "enum"
              },
              "SegmentationUpid": "string",
              "SegmentationTypeId": "integer",
              "SegmentsExpected": "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).
"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.

{
    "ChannelId": "string",
    "Creates": {
        "ScheduleActions": [
            {
                "ScheduleActionStartSettings": {
                    "FixedModeScheduleActionStartSettings": {
                        "Time": "string"
                    }
                },
                "ActionName": "string",
                "ScheduleActionSettings": {
                    "Scte35ReturnToNetworkSettings": {
                    }
                }
            }
        ]
    }
}

Example

This example of a request creates a return-to-network with a UTC start time of 20:42:19.
JSON for Create Actions

```json
{
    "ChannelId": "999999",
    "Creates": {
        "ScheduleActions": [
            {
                "ScheduleActionStartSettings": {
                    "FixedModeScheduleActionStartSettings": {
                        "Time": "2018-05-21T20:42:19.000Z"
                    }
                },
                "ActionName": "end_adavail_3708",
                "ScheduleActionSettings": {
                    "Scte35ReturnToNetworkSettings": {
                        "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.

```json
{
    "ChannelId": "string",
    "Creates": {
        "ScheduleActions": [
            {
                "ScheduleActionStartSettings": {
                    "FixedModeScheduleActionStartSettings": {
                        "Time": "string"
                    }
                },
                "ActionName": "string",
                "ScheduleActionSettings": {
                    "HlsTimedMetadataSettings": {
                        "Id3": "string"
                    }
                }
            }
        ]
    }
}
```

**Example**

This example of a request creates ID3 metadata to be inserted at 13:35:59 UTC.

```json
{
    "ChannelId": "999999",
    "Creates": {
        "ScheduleActions": [
            {
                "ScheduleActionStartSettings": {
                    "FixedModeScheduleActionStartSettings": {
                        "Time": "2019-01-02T13:35:59Z"
                    }
                },
            }
        ]
    }
}
```
"ActionName": "id3_metadata.2019-01-02T13:35:59Z",
"ScheduleActionSettings": {
  "HlsTimedMetadataSettings": {
    "Id3": "SUQzBAAAAAFVRYWFgAAALAAABIZWxsbyBDB3JzZA=="
  }
}
}

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": "string"
          }
        },
        "ActionName": "string",
        "ScheduleActionSettings": {
          "HlsId3SegmentTaggingSettings": {
            "Tag": "string"
          }
        }
      }
    ]
  }
}
```
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.

```
{
   "ChannelId": "8545690",
   "Creates": {
      "ScheduleActions": [
         {
            "ScheduleActionStartSettings": {
               "FixedModeScheduleActionStartSettings": {
                  "Time": "2019-03-10T20:42:19Z"
               }
            },
            "ActionName": "pause_pipeline_0_now",
            "ScheduleActionSettings": {
               "PauseStateSettings": {
                  "Pipelines": {
                     {
                        "PipelineId": "PIPELINE_0"
                     }
                  }
               }
            }
         }
      ]
   }
}
```

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 as 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 that the `Pipelines` array is empty. MediaLive interprets this empty array as "set all pipelines to unpaused".

```json
{
  "ChannelId": "999999",
  "Creates": {
    "ScheduleActions": [
      {
        "ScheduleActionStartSettings": {
          "FixedModeScheduleActionStartSettings": {
            "Time": "2018-05-21T20:52:00Z"
          }
        },
        "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.

```json
{
  "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
        }
      }
    }
  ]
}
```
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. 195).

```
{
  "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 executes 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.

```
{
  "ActionName": "SpliceInsert-02",
  "ScheduleActionStartSettings": {
    "FixedModeScheduleActionStartSettings": {
      "Time": "2018-11-05T16:30:45.000Z"
    }
  }
}
```
"ChannelId": "999999",
"Creates": {
"ScheduleActions": [
{
"ScheduleActionStartSettings": {
"FixedModeScheduleActionStartSettings": {
"Time": "2018-05-21T20:42:19.000Z"
}
},
"ActionName": "overlay-21",
"ScheduleActionSettings": {
"StaticImageActivateSettings": {
"Image": {
"PasswordParam": "imagespassword",
"Uri": "s3ssl://banners/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 executed in the channel
- Actions that have been executed 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://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": {  
        "FixedModeScheduleActionStartSettings": {  
          "Time": "2018-05-21T20:42:04.000Z"  
        }  
      }  
    }  
  ]  
}
```
"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://logos/corporate/high-res.bmp",
      "Username": "medialiveoperator"
    },
    "ImageY": 300,
    "FadeIn": 1500,
    "ImageX": 200,
    "Width": 800,
    "Opacity": 60,
    "Layer": 1,
    "Height": 900
  }
}
}
Starting, Stopping, and Pausing an AWS Elemental MediaLive Channel

After you create a channel, you can start it. The channel never starts automatically except when it is already running and attempts to recover from a failure.

You can stop a running channel at any time.

You can also pause one or both the pipelines in a channel by adding a Pause action to the schedule for the channel. For more information, see the section called “Types of Schedule Actions for Pause and Unpause” (p. 161).

For information about charges when a channel is running, see the section called “Pricing” (p. 6).

To start a channel

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

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 Channel Using the AWS Elemental MediaLive Console (p. 199)
- Monitoring a Multiplex Using the AWS Elemental MediaLive Console (p. 200)
- Monitoring a Channel or Multiplex Using Amazon CloudWatch Events (p. 202)
- Monitoring a Channel Using Amazon CloudWatch Logs (p. 205)
- Logging MediaLive API Calls with AWS CloudTrail (p. 207)

Monitoring 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)

2. In the navigation pane, choose Channels. (For information about the buttons on the page, see the section called “Editing a Channel” (p. 121), Starting, Stopping, and Pausing a Channel (p. 198), and the section called “Creating a Channel by Cloning” (p. 120).)
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 the section called “Pricing” (p. 6).
   - Recovering: One or both pipelines in the channel failed, but MediaLive is restarting it.
   - Running.
   - Starting
   - Stopping
   - Updating: You modified a channel, and MediaLive is updating the channel information.
4. To view more details about a channel, choose the name of that channel. The Channel details page appears.

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.

**Viewing Alerts**

AWS Elemental 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, AWS Elemental MediaLive automatically clears the alert.

If you stop a channel, alerts always automatically clear.

**Monitoring a Multiplex Using the AWS Elemental MediaLive Console**

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

**To monitor activity on a multiplex (AWS Elemental MediaLive console)**

1. EMLconsole_para
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 the section called "Pricing" (p. 6).
   - Recovering: One or both pipelines in the multiplex failed, but MediaLive is restarting it.
   - Running
   - Starting
   - Stopping
   - Updating: You modified a multiplex, and MediaLive is updating the multiplex information.
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. 296).

#### 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. 291).

#### 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 datapoints. Therefore, a window of 1 hour shows a datapoint every minute, for example. A window of 1 day shows a datapoint 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

AWS Elemental 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, AWS Elemental 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. 319).

---

**Monitoring a Channel or Multiplex Using Amazon CloudWatch Events**

AWS Elemental 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.

**Topics**

- Option 1: Send all MediaLive Events to an Email Address (p. 202)
- Option 2: Send Events for Specific Channels to an Email Address (p. 204)

**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)**

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)**

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:

```json
{
  "source": [  
    "aws.medialive"
  ]
}
```
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. 202), 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": [  
    ]
}
```

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": [  
]
```
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 a Channel or Multiplex Using CloudWatch” (p. 202)) does not provide enough information to resolve an issue on the channel.

Channel 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.

You should not automate any processing based on the wording in 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.)

There is a cost for channel logging. Logging of MediaLive channel activity forms part of your charges for Amazon CloudWatch Logs. See Amazon CloudWatch Pricing.

Topics
- Enabling Channel Logs (p. 205)
- Working with Logs (p. 206)

Enabling Channel Logs

You enable the capture of logging information 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.

To enable a channel 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. 121) (choose any idle channel), display the Channel and input details page (p. 72), and choose the Update button. When the role is updated in one channel, the change applies to all channels.
3. To enable logging in a new channel, set up logging during creation (p. 69). To enable logging in an existing channel, edit the channel (p. 121); 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. 80).
4. You or an administrator can also go into CloudWatch Logs and set an expiry date for the logs.

Disabling Channel Logs

You disable the capture of 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 MediaLive 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.

The log group is always the following: **ElementalMediaLive**.

The log stream is named after the ARN/pipeline. For example, `arn_aws_medialive_us-west-2_111122223333_channel_5106412_0`, where 5106412 is the channel ID and 0 is the pipeline.

Content of EML Logs

The MediaLive logs are in JSON format:

```json
{
    "encoder_pipeline": 0,
    "severity": "I",
    "timestamp": "2018-05-21T16:36:41.650318",
    "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. 72) 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. 206).
- **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

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.

<table>
<thead>
<tr>
<th>Level</th>
<th>Debug Messages</th>
<th>Info Messages</th>
<th>Warning Messages</th>
<th>Critical Messages</th>
<th>Fatal Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEBUG</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>INFO</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
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, a service that provides a record of actions taken by a user, role, or an AWS service in MediaLive. 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:

```json
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::111122223333:user/santosp",
    "accountId": "111122223333",
    "accessKeyId": "AKIAOSPONDNYEXAMPLE",
    "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"
    ],
    "roleArn": "MediaLiveAccessRole",
    "requestId": "1111aaaa-9604-4459-a160-46a28ae166",
    "name": "live_studio_feed",
    "type": "RTP_PUSH"
  },
  "responseElements": {
    "input": {
      "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"}
AWS Elemental MediaLive Features

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. 69) and MediaLive Schedule (p. 159).

Topics
- Audio Rendition Groups for HLS (p. 210)
- Automatic input failover  (p. 218)
- Working with Captions (p. 224)
- Channel class—implementing pipeline redundancy (p. 243)
- Color Space Handling in AWS Elemental MediaLive (p. 246)
- Dynamic Inputs (p. 254)
- Working with ID3 Metadata (p. 255)
- Working with ID3 Segment Tags (p. 258)
- Working with Image Overlays (p. 260)
- Input Clipping (p. 262)
- Input Switching in AWS Elemental MediaLive (p. 262)
- Customizing the paths inside HLS manifests (p. 280)
- Redundant HLS manifests (p. 285)
- Multiplex and MPTS in AWS Elemental MediaLive (p. 290)
- Converting Nielsen Watermarks to ID3 (p. 297)
- Implementing resiliency in the channel (p. 298)
- SCTE-35 Message Processing (p. 299)
- Tagging AWS Elemental MediaLive Resources (p. 319)
- Timecode Configuration (p. 321)
- Setting the Rate Control Mode (p. 323)
- Reference: Supported Captions (p. 325)
- Reference: Supported Containers and Codecs (p. 336)
- Reference: Identifiers for Variable Data (p. 343)

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.
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.

**Note**

The information in this section assumes that you are familiar with the general steps for creating a channel, as described in Creating a Channel from Scratch. 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. 89).

**Topics**

- About Audio Rendition Groups (p. 211)
- Creating an Output with an Audio Rendition Group (p. 212)
- Sample Manifest (p. 217)

### 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.](http://example.com)

#### 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.
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. The encodes and associations that you create are the following:

- 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 with the audio rendition group, you assign the audio group ID to that video output.
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 the 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. 214) 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.

![Diagram of video and audio encodes and rendition groups]
Rules for Video and Audio in Rendition Groups

- 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.
Creating a Rendition Group

<table>
<thead>
<tr>
<th>Value for a given audio rendition</th>
<th>Client player behavior</th>
<th>Representation in HLS Manifest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alternate Audio, Auto Select, Default</td>
<td>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.</td>
<td>EXT-X-MEDIA with DEFAULT=YES, AUTOSELECT=YES</td>
</tr>
<tr>
<td>Alternate Audio, Auto Select, Not Default</td>
<td>The client player might select this audio rendition. Any number of renditions in the rendition group can be set this way.</td>
<td>EXT-X-MEDIA with DEFAULT=NO, AUTOSELECT=YES</td>
</tr>
<tr>
<td>Alternate Audio, not Auto Select</td>
<td>The client player should never select this audio rendition. Any number of renditions in the rendition group can be set this way.</td>
<td>EXT-X-MEDIA with DEFAULT=NO, AUTOSELECT=NO</td>
</tr>
<tr>
<td>Audio-Only Variant Stream</td>
<td>The client can play back this audio-only rendition instead of video, in low-bandwidth scenarios.</td>
<td>EXT-X-STREAM-INF</td>
</tr>
</tbody>
</table>

**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

Follow this procedure for each video 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 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. 100).
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. 214).
   - 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.
- 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.
Automatic input failover

When you set up the inputs for a channel, you can set up two inputs as an input failover pair (or 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.

The input pair provides content to the same pipeline in 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. 243), which is another form of channel resiliency. You might decide to implement one or both of these features.

**Topics**

- About automatic input failover (p. 219)
- Setting up automatic input failover with MediaConnect inputs (p. 219)
- Setting up automatic input failover with RTMP and RTP inputs (p. 221)
- Changing the roles of the failover pair (p. 222)
- Starting the channel (p. 222)
- Automatic input failover and input switching (p. 223)
About automatic input failover

Eligible inputs

You can set up two push inputs as a failover pair. Push inputs are always live inputs. See the section called "Supported Input Types and Upstream Systems" (p. 336) for a list of push inputs in MediaLive.

Single-input channels

In a workflow with only one content source, you create set up for automatic input failover if the content is a live, push source. You create two inputs that connect to that single source, then set up the two inputs in a failover pair.

If your workflow has only one content source but it is a pull source, you can't set up for automatic input failover.

Multiple-input channels

In a workflow with multiple inputs (meaning that you plan to switch among different inputs), you can set up for automatic input failover if you have only one live content source and it is a push source. You create two inputs that connect to the same live source, then set up those two inputs in a failover pair.

The other inputs must all be file inputs. You can't attach another live input because you've used up your live input quota on the failover pair.

Identical sources

The two inputs in the pair must contain identical content—the video, the audio, and captions must have identical characteristics.

Coordination with your upstream system

Contact the upstream system to determine if they are capable of sending your two source streams for the relevant input. If the upstream system can't provide two sources, you can't implement automatic input failover.

Automated input failover and pipeline redundancy

You can implement both automatic input failover and pipeline redundancy (p. 243) (standard channel). If you implement both features, the source requirements are different: you need four sources from the upstream system—two for each input, and two for each pipeline.

Implementing both features provides more resiliency:

- With automatic input failover, when an input fails (the Availability Zone fails), only one flow to the pipeline fails. MediaLive switches to the other flow.
- With a standard channel, when a pipeline fails, output continues on the other pipeline.

The following procedures describe how to set up for both scenarios—automatic input failover in a single-pipeline channel, and automatic input failover in a standard channel.

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. 129) and creating a channel (p. 69).
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

- Follow the procedure in the section called “Creating a MediaConnect push input” (p. 129), 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. 223).

To create the flows in MediaConnect in a single-pipeline channel

- Follow the procedure in the section called “Creating a MediaConnect push input” (p. 129), with the following note:

  Make sure that you set up the flows in different Availability Zones. The two flows you create for use with MediaLive must always be in different Availability Zones. The two Availability Zones provide two paths to MediaLive. If one Availability Zone fails, one input fails. MediaLive switches to the other input, which is in a different Availability Zone.

To create the inputs for the input failover pair

1. Follow the procedure in the section called “Creating a MediaConnect push input” (p. 129) 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.
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. Complete the **Automatic input failover settings** section:
   - For **Secondary input**, choose the secondary input. (When you do this, this input is labeled as **Secondary** in the list of attachments.)
   - 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.

### Setting up automatic input failover with RTMP and RTP inputs

To use RTMP 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. 127) and **creating a channel** (p. 69).

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 redundancy.
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 *MediaLive Input* (p. 126) 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. Complete the *Automatic input failover settings* section:
   - For *Secondary input*, choose the secondary input. (When you do this, this input is labeled as *Secondary* in the list of attachments.)
   - 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.

**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:
Automatic input failover and input switching

- 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 “Switching within the input failover pair” (p. 224).

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.

Automatic input failover and input switching

When you implement automatic input failover, you can still implement input switching.

With automatic input failover, your deployment contains an input failover pair that uses up your quota of live inputs for the channel. You can't attach more live inputs to the channel.

Therefore, in terms of changing content in the channel, you can switch between any of the file inputs and the single live input pair. There are also special cases for switching between the primary and secondary input.

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. 163).

Switching inputs

You can set up the schedule to switch between inputs as follows:
- From a file input to another file input.
- From a file input to either input in the failover pair.
- From the primary input or secondary input to a file input.
Switching within the input failover pair

You can switch between the inputs in the failover pair only if the Input preference is set to EQUAL_INPUT_PREFERENCE.

Keep in mind that the content in the two live inputs 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**

- Create an input switch action in the schedule in the usual way.

  Set up the input to switch to the other input, and set the Start Type to Immediate.

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. 69). It also assumes that you have started creating a channel, including associating an input with the channel.

**Topics**

- Supported Features (p. 224)
- Typical Scenarios (p. 226)
- Setting Up for Captions (p. 228)
- Examples (p. 236)

Supported Features

This section provides information on the different features of captions that AWS Elemental MediaLive supports.

**Topics**

- Supported Formats (p. 225)
- Format Support by Output Container (p. 225)
- Support for Languages (p. 225)
• Support for Font Styles in Output Captions (p. 225)

Supported Formats

AWS Elemental MediaLive supports specific formats in inputs and specific formats in outputs. See the section called “General Information About Supported Formats” (p. 325) 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, if your input container is an MP4 container and your output is HLS, and you want to include Web-VTT captions in the HLS output, you can do so if the MP4 container holds 608 embedded captions. But you can't include Web-VTT captions if 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. 325).

Support for 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 – For a Teletext source, if you specify Teletext as 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.
• 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.

**Font Style Options**

<table>
<thead>
<tr>
<th>Source Captions</th>
<th>Output Captions</th>
<th>Options for Font Style</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARIB</td>
<td>ARIB</td>
<td>None. The font styles in the input are automatically passed through in the output.</td>
</tr>
<tr>
<td>Any supported captions format</td>
<td>Burn-in</td>
<td>You can specify font styles in the output. If you don't specify styles, the AWS Elemental MediaLive defaults are used.</td>
</tr>
<tr>
<td>DVB-Sub</td>
<td>DVB-Sub</td>
<td>None. The font styles in the input are automatically passed through in the output.</td>
</tr>
<tr>
<td>Any supported captions format</td>
<td>DVB-Sub</td>
<td>You can specify font styles in the output. If you don't specify styles, the AWS Elemental MediaLive defaults are used.</td>
</tr>
<tr>
<td>Teletext</td>
<td>Teletext</td>
<td>None. The font styles in the input are automatically passed through in the output.</td>
</tr>
<tr>
<td>An Embedded Combination (Embedded, Embedded +SCTE-20, SCTE-20+Embedded)</td>
<td>TTML</td>
<td>The font information in the source can be copied to the output, or you can let the downstream player determine the font style.</td>
</tr>
<tr>
<td>Teletext</td>
<td>TTML</td>
<td>The font information in the source can be copied to the output, or you can let the downstream player determine the font style.</td>
</tr>
<tr>
<td>Any Other</td>
<td>Any Other</td>
<td>No control: the font style is always determined by the downstream player.</td>
</tr>
</tbody>
</table>

**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** (p. 227)
- **Use Case B: One Input Format Converted to One Different Format in One Output** (p. 227)
- **Use Case C: One Input Format Converted to Different Formats, One Format for Each Output** (p. 227)
- **Use Case D: One Captions Output Shared by Multiple Video Encodes** (p. 228)
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 Web-VTT.
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 (p. 229)
- Step 2: Plan Captions for the Outputs (p. 232)
- Step 3: Match Formats to Categories (p. 233)
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.
3. Determine how many captions selectors to create in the input in the channel, using the following guidance:

- **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. 230).
- **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.
- **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. 231).
- **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.
- **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. 230), Embedded (p. 230), or Teletext (p. 231).
6. Create more captions selectors, as required.
Information for DVB-Sub or SCTE-27

DVB-Sub and SCTE-27 formats are supported only in TS inputs. You must specify the location of the captions by completing the PID or Language code fields in one of these ways.

<table>
<thead>
<tr>
<th>PID</th>
<th>Language Code</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specified</td>
<td>Blank</td>
<td>Extracts captions from the specified PID.</td>
</tr>
<tr>
<td>Blank</td>
<td>Specified</td>
<td>Extracts the specified language, whichever PID that happens to be in.</td>
</tr>
<tr>
<td>Specified</td>
<td>Specified</td>
<td>Extracts captions from that PID; the language is informational.</td>
</tr>
<tr>
<td>Blank</td>
<td>Blank</td>
<td>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.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>EIA-608 in Source</th>
<th>CEA-708 in Source</th>
<th>Convert Field</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td><strong>Upconvert</strong></td>
<td>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.</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td><strong>Disabled</strong></td>
<td>Original EIA-608 is preserved.</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td><strong>Upconvert</strong></td>
<td>Original CEA-708 is preserved.</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td><strong>Disabled</strong></td>
<td>Original CEA-708 is preserved.</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td><strong>Upconvert</strong></td>
<td>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.</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td><strong>Disabled</strong></td>
<td>Original EIA-608 is preserved and original CEA-708 is preserved.</td>
</tr>
</tbody>
</table>

• **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. Teletext passthrough is supported only in a TS output.
• 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.

Teletext is a form of data that can contain several types of information, not just captions. MediaLive supports only extraction of the entire Teletext input for passthrough to the output.

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), 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.

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. 229), 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 Web-VTT 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. 325).
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 Web-VTT captions in Czech
- HLS output with Web-VTT 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 Web-VTT 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. 325).
- 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. 225).

**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. 327).

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. 233)
- the section called "Sidecar Captions and SMPTE-TT in Microsoft Smooth" (p. 234)

**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 in step 1 (p. 229). 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. 235).

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. 236).

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. 235) or PIDs for DVB-Sub output (p. 235).

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. 327).

You set up each captions asset in its own output within the output group.

To set up the 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, Web-VTT Czech. Or accept the default (which is automatically generated).

7. For Captions selector name, enter the name of the captions selector that you created in step 1 (p. 229). 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.

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 both 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, 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.

<table>
<thead>
<tr>
<th>ARIB Captions PID Control</th>
<th>ARIB Captions PID</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto</td>
<td>Ignore</td>
<td>A PID is automatically assigned during encoding. This value could be any number.</td>
</tr>
<tr>
<td>Use Configured</td>
<td>Enter a decimal or hexadecimal</td>
<td>This PID is used for the captions.</td>
</tr>
</tbody>
</table>

**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.

**Examples**

The following examples describe how to implement the use cases from the section called “Typical Scenarios” (p. 226).

**Topics**

- **Use Case A: One Input Format to One Output and Not Converted** (p. 236)
- **Use Case B: One Input Format Converted to One Different Output Format** (p. 237)
- **Use Case C: One Input Format Converted to Different Formats, One Format for Each Output** (p. 238)
- **Use Case D: One Captions Output Shared by Multiple Video Encodes** (p. 240)

**Use Case A: One Input Format to One Output and Not Converted**

This example shows how to implement the first use case (p. 227) 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. 227) 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. 227) 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 Web-VTT.
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**: Web-VTT.
  • **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**: Web-VTT
  • **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.

**Setup with Embedded or Object-style Captions**

This example shows how to implement the fourth use case (p. 227) (p. 228) 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.

Channel class—implementing pipeline redundancy

You can set up a channel with two pipelines, to provide resiliency within the channel pipeline. You should decide whether to implement pipeline redundancy when you plan the workflow (p. 57).

You set up for pipeline redundancy by setting the channel class:

• Standard channel
• Single-pipeline channel

You set up for pipeline redundancy by setting up the channel as a standard channel. The channel has two pipelines—pipelines 0 and 1. Both pipelines ingest, process the source content, and produce output. The downstream system handles output from one of the pipelines and ignores the other. 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.

If you don’t want pipeline redundancy, you set up the channel as a single-pipeline channel. If that single pipeline fails, output to the downstream system stops.

Note
Before you decide to implement pipeline redundancy, you should read about automatic input failover (p. 218), which is another form of channel resiliency. You might decide to implement one or both of these features.

Topics
• Guidelines for channel class (p. 244)
• Implementing pipeline redundancy in a new channel (p. 244)
• Changing pipeline redundancy in an existing channel (p. 246)
Guidelines for channel class

Before you decide to set up a standard channel, in order to implement pipeline redundancy, you should be aware of all the requirements:

- Contact the downstream system to determine if the downstream system can handle two sets of identical outputs from MediaLive and to switch as required.

  If the downstream system doesn't have this ability, there is not much advantage to setting up as a standard channel.

- Contact the upstream system to determine if they can send you two source streams for each input. If the upstream system can't provide two sources, you can't set up as a standard channel.

  Furthermore, if you have more than one upstream system for the channel, every upstream system must be capable of providing two sources. If every upstream system can't do this, you can't set up as a standard channel.

- 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 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.

- If you decide not to implement pipeline redundancy, you should decide if you want to leave open the option of implementing it later on.

  Or if you do decide to implement pipeline redundancy, you should decide if you want to leave open the option of removing it later on.

  In both cases, you must make sure to set up the input class appropriately. See the section called “About input classes” (p. 245) and make a note of the input class that you want to use.

Implementing pipeline redundancy in a new channel

To implement pipeline redundancy on a new channel, you must set up the inputs and the channel in a specific way.

To implement pipeline redundancy

1. Decide if you will set up the inputs as standard inputs or single-pipeline inputs. For more information, see the section called “About input classes” (p. 245).

   You must set all the inputs to the same class.

2. Create the input or inputs that you identified when you planned the workflow (p. 57). For more information about creating inputs, see MediaLive Input (p. 126).

   Make sure that you set the Channel and input class field in the input to the class that you identified:

   - If you choose STANDARD, two source fields appear.
   - If you choose SINGLE-PIPELINE, one source field appears.

   (This field doesn't appear for RTP inputs—these inputs are always set up with two endpoints.)

3. Create the channel with these special notes:
• On the **Channel and input details** page, for **Channel class**, choose a valid value. See the table below.

• On the **Attach input** page, make sure that you attach valid inputs. See the table below.

### About input classes

If you decide to implement input resiliency, you must decide which input class to use for the input. There are two input classes—standard and single-pipeline.

• With a standard input, you can specify either one source or two source URLs.

• With a single-pipeline input, you can specify only one source URL.

You can set up the combination of channel class and input class in one of the following ways:

• Set up a standard channel with standard inputs

• Set up a single-pipeline channel with standard inputs and upgrade potential

• Set up a single-pipeline channel with single-pipeline inputs

### Standard channel with standard inputs

With this setup, the channel has two pipelines. This setup supports pipeline redundancy.

Set up the input with two sources—you must specify two source URLs in the input, and the upstream system must provide two sources. If one pipeline fails, the channel continues to produce output from the other pipeline.

This setup lets you later downgrade the channel to a single-pipeline channel with minimum disruption. To downgrade, you stop the channel, change the channel class, and restart. At restart, MediaLive has one pipeline and it simply ignores the second source. For more information about downgrading, see the section called “Updating Channel Class” (p. 122).

### Single-pipeline channel with standard inputs and upgrade potential

With this setup, the channel has one pipeline. This setup doesn't support pipeline redundancy, but you can easily set it up later.

Even though the channel has only one pipeline, you set up the input with two URLs. But you set up the upstream system to provide only one source. The single source provides content to the single pipeline. If the single pipeline fails, the channel stops producing output.

This setup supports upgrade to a standard channel with minimum disruption. To upgrade, you stop the channel, change the channel class, and restart. You also start providing content to the second URL, which previously had been dormant. At restart, MediaLive has two pipelines and two sources, one for each pipeline. For more information about upgrading, see the section called “Updating Channel Class” (p. 122).

### Single-pipeline channel with single-pipeline inputs

With this setup, the channel has one pipeline. This setup doesn't support pipeline redundancy.

You set up the input with one URL. You set up the upstream system to provide only one source. The single source provides content to the single pipeline. If the single pipeline fails, the channel stops producing output.

You can upgrade this channel to a standard channel, but the procedure is disruptive. You must stop the channel, change the channel class, edit each input to specify two URLs, and restart the channel. For more information about upgrading, see the section called “Updating Channel Class” (p. 122).
We recommend that you set up in this way only if you are sure you won't want to upgrade the channel to a standard channel. You might decide that you don't need to consider upgrade because you are providing channel resiliency through automatic input failover (p. 218).

Valid combinations of channel class and input class

The following table describes all the value combinations of channel class and input class. Read each row as follows. The first cell specifies a channel class. The second cell specifies the classes of inputs you can attach to the channel. The third cell describes the results.

<table>
<thead>
<tr>
<th>Channel class</th>
<th>Class of the inputs</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>STANDARD</td>
<td>All inputs are STANDARD</td>
<td>MediaLive implements pipeline redundancy by creating two pipelines in the channel.</td>
</tr>
<tr>
<td>SINGLE-INPUT</td>
<td>All inputs are SINGLE-INPUT</td>
<td>MediaLive doesn't implement pipeline redundancy. The channel has one pipeline.</td>
</tr>
<tr>
<td></td>
<td>All inputs are STANDARD</td>
<td>MediaLive doesn't implement pipeline redundancy. The channel has one pipeline. However, you have set up for easy upgrade to a standard channel.</td>
</tr>
<tr>
<td></td>
<td>Some inputs are STANDARD and some are SINGLE-PIPELINE</td>
<td>MediaLive doesn't implement pipeline redundancy. The channel has one pipeline. You haven't set up for easy upgrade.</td>
</tr>
</tbody>
</table>

Changing pipeline redundancy in an existing channel

To enable or disable pipeline redundancy on an existing channel, you must update the channel class. For information, see the section called “Updating Channel Class” (p. 122).

Color Space Handling in AWS Elemental MediaLive

The input video may include three sets of metadata that relate to video color. When this metadata information is present, it enhances the display of the video content on the downstream player.

The three sets of metadata are the following:

- Color space that specifies the range of pixel colors that apply to the content. The content is said to be marked for a color space.
  
  MediaLive supports two SDR color spaces: the 601 color space or the 709 color space.
  
  MediaLive supports two HDR color spaces: the HDR10 color space or the HLG color space.
- Brightness function used for the pixel. Also known as gamma tables, lookup tables (LUT), Electro-optical Transfer Function (EOTF), and transfer function.
- Display metadata.
MediaLive provides two features that relate to color metadata handling:

- On the input side, MediaLive supports the ability to pass through any color space metadata to the output, or to correct the color space metadata and pass that corrected metadata to the output.

  By default, MediaLive passes through (without correction) any color space metadata that is present. It passes through both SDR and HDR metadata.

- On the output side, MediaLive supports the ability to convert the color space metadata that is in the video source to another color space. For example, you can convert source content that is 601 color space so that it is 709 color space in the output.

  By default, MediaLive doesn't convert the color space metadata. It passes through the source color space.

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. 69). This section provides supplemental information about configuring for color space handling.

Topics
- General Information about Color Space (p. 247)
- Configuring the Handling of the Input (p. 248)
- Configuring the Handling of the Output (p. 249)

General Information about Color Space

Following is some general information about color space.

Color Space Standards

The applicable standards for the three sets of color data are different for different types of color space.

<table>
<thead>
<tr>
<th>MediaLive Term for the Color Space</th>
<th>Complies with This Color Space Standard</th>
<th>Complies with This Brightness Function Standard</th>
<th>Complies with This Standard for Display Metadata</th>
</tr>
</thead>
<tbody>
<tr>
<td>rec601</td>
<td>SDR rec. 601</td>
<td>BT.1886</td>
<td>Not applicable</td>
</tr>
<tr>
<td>rec709</td>
<td>SDR rec. 709</td>
<td>BT.1886</td>
<td>Not applicable</td>
</tr>
<tr>
<td>HDR10</td>
<td>HDR10</td>
<td>SMPTE ST 2084 (PQ)</td>
<td>SMPTE ST 2086</td>
</tr>
<tr>
<td>HLG 2020</td>
<td>HDR10</td>
<td>HLG rec. 2020</td>
<td>Not supported</td>
</tr>
</tbody>
</table>

How Color Space and Video Resolution Work Together

Color space refers to the range of color. MediaLive supports SDR (standard dynamic range) and HDR (high dynamic range) color spaces.

Resolution refers to the video pixel count. MediaLive supports standard definition (SD) resolutions, high definition (HD) resolutions, and ultra-high definition (UHD) resolutions. For UHD, it supports up to 4K.

In terms of combinations of color space and resolution, the following rules apply:
• SDR color space can be associated with SD, HD, and UHD video.
• HDR color space can be associated with HD or UHD video. It can't be associated with SD video.

Configuring the Handling of the Input

You should identify the color spaces that are in the video source, and decide how you want to handle the metadata.

To identify your requirements

1. Contact the content provider to determine what color space or color spaces apply to the video sources.

   Any source might be marked for a color space, or it might be marked as "unknown," or it might not be marked (unmarked). Content might be any combination of marked, unknown, and unmarked.

2. Decide if you want to pass through or correct the color space marking in the input.

   For the correction option, the following rules apply:

   • This option doesn't convert the color space metadata. It inserts metadata that is missing, or it changes the metadata for incorrectly marked content.
   • This option applies only to the SDR color spaces. (You can't correct HDR content, but typically HDR content is correctly marked.)
   • Any correction that you configure applies to all inputs for the entire life of the channel.
   • Conversion of color space (which you can configure on the channel outputs (p. 249)) applies only to marked content. So if you use correction to insert missing metadata, you can increase the percentage of the content that gets converted in the output.
   • Only correct the color space if you are sure that all the unmarked portions use the color space that you choose.

To set up the input

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. For details, see the following table.

In this table, the first two columns specify the possible combinations of values in the two color space fields. The third field specifies the result for each combination. The last column specifies which input color spaces the combination is valid for.

<table>
<thead>
<tr>
<th>Color Space</th>
<th>Color Space Usage</th>
<th>Result</th>
<th>Valid for These Color Spaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>Follow</td>
<td>This field is ignored.</td>
<td>MediaLive doesn't change the color space metadata.</td>
<td>All sources.</td>
</tr>
<tr>
<td>A specific SDR color space</td>
<td>Force</td>
<td>MediaLive marks all the content as using the specified color space.</td>
<td>SDR color spaces. Marking content as using an SDR color space when it actually</td>
</tr>
</tbody>
</table>
Configuring the Handling of the Output

For each output, decide if you want a different color space in the output than is in the source. If you do, you must configure the output to convert the existing color space. MediaLive supports the following conversions:

- Converting from one SDR color space to the other SDR color space.
- Converting from one HDR color space to the other HDR color space.
- Converting from an SDR color space to an HDR color space. This conversion fits the SDR color space into the larger HDR color space and maps the pixels to new code values that represent the same color. It doesn't actually make the existing color richer.

Typically, you don't choose to convert an SDR to HDR. Instead, this conversion occurs 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 just stated the conversion has no effect on the color of the SDR content.

Topics

- Set Up the Output if the Video Resolution Is SD (p. 249)
- Set Up the Output if the Video Resolution Is HD or UHD (p. 251)

Set Up the Output if the Video Resolution Is SD

Follow this procedure to configure color space handling in an output that you plan to configure for an SD video resolution.

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 set up an SD 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. Complete the Width and Height fields to specify a valid SD resolution.
4. For Codec settings, choose H264 (AVC) or H265 (HEVC).
5. Choose **Codec details**, and then complete the **Profile**, **Tier** (for H265 only), and **Level** fields.

6. For **Color space**, choose **Color space settings**, and then choose the appropriate option to remove, pass through, or convert the source color space metadata. For information about the options, see the following tables.

In this table, the first column lists the possible combinations of resolution and color space in the input. The second column specifies the possible color space handling in the output. The third column specifies the value in **Color correction** to choose.

<table>
<thead>
<tr>
<th>Input Resolution and Color Space</th>
<th>Valid Handling of Color Space</th>
<th>Option on Console</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD with SDR</td>
<td>Include SDR color space</td>
<td>Color space passthrough or Rec601 or Rec709</td>
</tr>
<tr>
<td></td>
<td>No color space metadata</td>
<td>Don't include</td>
</tr>
<tr>
<td>HD with SDR</td>
<td>Include SDR color space</td>
<td>Color space passthrough or Rec601 or Rec709</td>
</tr>
<tr>
<td></td>
<td>No color space metadata</td>
<td>Don't include</td>
</tr>
<tr>
<td>HD with HDR</td>
<td>No color space metadata</td>
<td>Don't include</td>
</tr>
<tr>
<td>SD or HD with no color space metadata</td>
<td>No color space metadata</td>
<td>Don't include</td>
</tr>
</tbody>
</table>

This table specifies which options are valid for which source color space, and describes the handling for each option.

<table>
<thead>
<tr>
<th>Option</th>
<th>Valid for These Source Color Spaces</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Don't include</td>
<td>All color spaces and no color space</td>
<td>Remove the metadata that is in the source video. The output won't contain any color space metadata, brightness metadata, or display metadata.</td>
</tr>
<tr>
<td>Color space passthrough</td>
<td>All sources</td>
<td>Pass through the metadata from the source video. Passes through the color space and the brightness metadata. Any unmarked or unknown content in the source remains unmarked or unknown in the output.</td>
</tr>
</tbody>
</table>
## Configuring Output

<table>
<thead>
<tr>
<th>Option</th>
<th>Valid for These Source Color Spaces</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rec601</td>
<td>SDR 709</td>
<td>Convert content that is marked as 709 color space to 601, and pass through any brightness metadata from the source video. Any unmarked or unknown content in the source remains unmarked or unknown in the output. Any content that is marked as 601, HDR10, or HLG in the source remains as is, with no changes.</td>
</tr>
<tr>
<td>Rec709</td>
<td>SDR 601</td>
<td>Convert content that is marked as 601 color space to 709, and pass through any brightness metadata from the source video. Any unmarked or unknown content in the source remains unmarked or unknown in the output. Any content that is marked as 709, HDR10, or HLG in the source remains as is, with no changes.</td>
</tr>
</tbody>
</table>

### Set Up the Output if the Video Resolution Is HD or UHD

Follow this procedure to configure color space handling in an output that you plan to configure for an HD video resolution.

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 set up HD 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. Complete the Width and Height fields to specify a valid HD or UHD resolution.
4. For Codec settings, choose H264 (AVC) or H265 (HEVC).
5. 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.
6. Choose **Color space**, and then for **Color space settings**, choose the appropriate option to remove, pass through, or convert the source color space metadata. For information about the options, see the tables after the last step of this procedure.

7. If you choose HDR10 for the **Color space settings**, you can optionally complete **Max CLL** and **Max FALL** to set brightness metadata. These fields are provided because MediaLive doesn’t support brightness for HLG (the color space in the source), but it does support brightness metadata for HDR10.

In this table, the first column lists the possible combinations of resolution and color space in the input. The second column specifies the possible color space handling in the output.

<table>
<thead>
<tr>
<th>Input Resolution and Color Space</th>
<th>Valid Handling of Color Space</th>
<th>Option on Console</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD with SDR</td>
<td>SDR</td>
<td>Color space passthrough or Rec601 or Rec709</td>
</tr>
<tr>
<td>SD with no color space metadata</td>
<td>No color space metadata</td>
<td>Don't include</td>
</tr>
<tr>
<td>HD with SDR</td>
<td>SDR</td>
<td>Color space passthrough or Rec601 or Rec709</td>
</tr>
<tr>
<td>HD with no color space metadata</td>
<td>No color space metadata</td>
<td>Don't include</td>
</tr>
<tr>
<td>HD with HDR</td>
<td>HDR</td>
<td>Color space passthrough or HDR10</td>
</tr>
<tr>
<td>HD with no color space metadata</td>
<td>No color space metadata</td>
<td>Don't include</td>
</tr>
</tbody>
</table>

This table specifies which options are valid for which source color space, and describes the handling for each option.

<table>
<thead>
<tr>
<th>Option</th>
<th>Valid for These Source Color Spaces</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Don't include</strong></td>
<td>All sources</td>
<td>Remove the metadata that is in the source video.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The output won't contain any color space metadata, brightness metadata, or display metadata.</td>
</tr>
<tr>
<td><strong>Color space passthrough</strong></td>
<td>All sources</td>
<td>Pass through the metadata from the source video.</td>
</tr>
<tr>
<td>Option</td>
<td>Valid for These Source Color Spaces</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Passes through the color space, the brightness metadata, and the display metadata (if the source is HDR10). Any unmarked or unknown content in the source remains unmarked or unknown in the output.</td>
</tr>
<tr>
<td>Rec601</td>
<td>SDR 709</td>
<td>Convert content that is marked as 709 color space to 601, and pass through any brightness metadata from the source video. Any unmarked or unknown content in the source remains unmarked or unknown in the output. Any content that is marked as 601, HDR10, or HLG in the source remains as is (no changes) in the output.</td>
</tr>
<tr>
<td>Rec709</td>
<td>SDR 601</td>
<td>Convert content that is marked as 601 color space to 709, and pass through any brightness metadata from the source video. Any unmarked or unknown content in the source remains unmarked or unknown in the output. Any content that is marked as 709, HDR10, or HLG in the source remains as is (no changes) in the output.</td>
</tr>
</tbody>
</table>
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

How Dynamic Inputs Work

To set up an input as a dynamic input, you include a variable as all or part of the URL of the file. For example:

`s3://$urlPath$

You attach the input to the channel when you create the channel.

You set up the schedule to include switch input actions to set up the ingest order for inputs.

For each schedule action that uses a dynamic input, you include a value for the variable portion of the URL. For example:

<table>
<thead>
<tr>
<th>Option</th>
<th>Valid for These Source Color Spaces</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDR10</td>
<td>HLG</td>
<td>This option appears only if you set the video codec to H265. Convert content that is marked as HLG, 602, or 709 color space to HDR10, and pass through any display metadata from the source video. This option also passes through any brightness metadata, unless you overwrite it by completing the Max Cll and Max Fall fields. Any unmarked or unknown content in the source remains unmarked or unknown in the output. Any content that is already marked as HDR10 in the source remains as is (no changes) in the output.</td>
</tr>
</tbody>
</table>
movies/my-movie.mp4

When the channel switches to this input, it resolves the URL:
s3://movies/my-movie.mp4

The next time you use this input in an input switch action, you might set up the action to resolve the URL to a different path:
s3://movies/my-movie-the-sequel.mp4

Setting Up Dynamic Inputs

For information about planning and creating dynamic inputs, see the section called “Input Switching” (p. 262).

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. 256) for the output type.

Topics
- Enabling ID3 Metadata (p. 255)
- Passing Through ID3 Metadata (p. 257)
- Inserting ID3 Metadata When Creating the Channel (p. 257)
- Inserting ID3 Metadata Using the Schedule (p. 258)

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. 256) 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. 256) 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. 256) 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. 256) 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. 257).

• 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. 258).

The eligibility of an output depends on the output group type, as shown in the following table.

<table>
<thead>
<tr>
<th>Type of Output Group</th>
<th>ID3 Metadata That Is Present in Input</th>
<th>ID3 Metadata That You Specify When Setting Up the Channel</th>
<th>ID3 Metadata That You Insert Using the Schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archive</td>
<td>Passed through</td>
<td>Not included in output</td>
<td>Included in output</td>
</tr>
<tr>
<td>HLS</td>
<td>Passed through</td>
<td>Included in output</td>
<td>Included in output</td>
</tr>
<tr>
<td>MediaPackage</td>
<td>Passed through</td>
<td>Not included in output</td>
<td>Included in output</td>
</tr>
<tr>
<td>UDP</td>
<td>Passed through</td>
<td>Included in output</td>
<td>Not included in output</td>
</tr>
</tbody>
</table>

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. 255).

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. 255).

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. 255).
2. Create actions in the schedule. For detailed information, see **MediaLive Schedule** (p. 159).

### 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, MediaLivestarts 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. 255). 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:
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 *MediaLive Schedule (p. 159).*

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. 159) 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 that you overlay on a video can be a .bmp, .png, or .tga file.

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 each image overlay with a start time and duration. You can insert the image overlay at any position on the video frame, as specified by x/y coordinates. You can configure with an opacity and with fade-in and fade-out.

The image is handled as follows:

- The image is overlaid on the underlying video pixel for pixel, without scaling.
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 four ways:
   - Amazon S3 bucket, using SSL. For example:
     s3ssl://company-overlays/sample_bucket/overlay.png
   - Amazon S3 bucket, without SSL. For example:
     s3://company-overlays/sample_bucket/overlay.png
   - 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 MediaLive Schedule (p. 159) and the section called "Creating Actions" (p. 163).
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 or AWS Elemental MediaStore. You can't clip an MP4 file that is stored on a remote server.

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.

**To set up a file input for input clipping**

1. Create the MP4 file input in the usual way. See the section called “Creating an MP4 pull input” (p. 131).
2. Attach the input to the channel in the usual way. See the section called “Step 2: Attach inputs” (p. 73).
3. 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. 163).

   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.

**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**

- Overview of Input Switching (p. 262)
- Rules and Limits for Input Switches  (p. 264)
- Types of Switches (p. 265)
- Setting Up for Input Switching (p. 266)
- Starting and Restarting a Channel That Has Multiple Inputs (p. 279)

**Overview of Input Switching**

You set up input switching in a channel in order to ingest the inputs in a multiple-input channel.
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.

**Rules and Limits for Input Switches**

This section describes the rules and limits that apply to input switches.

**Rules for Types of Inputs**

There are two restrictions on the different input types that can be in the "pool" of inputs attached to one channel:

- 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, including both RTP VPC inputs and RTMP VPC push inputs. But all those inputs must be in the same two Availability Zones.

For example:

- You can have both HLS 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.
Limits to Inputs in a Channel

Each channel that implements input switching can contain a specific number of inputs, as follows:

• A maximum 20 inputs.
• From 0 to 2 of those inputs can be live inputs.
• The remainder of the 20 inputs can be file inputs. File inputs are file-based video inputs.

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.

Types of Switches

There are several types of switches according to the start mode—the rule that triggers the switch.

Fixed, Immediate, and Follow Switches

There are three types of input switch start modes:

• Fixed – These input switches start at a specific UTC time.
• Immediate – The input switches start 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.
  Typically, you don't set up an immediate switch in advance. You enter an immediate switch only on an ad hoc basis, to perform a switch that was not planned when you set up the schedule.
• Follow – These input switches start when the previous input has ended (when MediaLive has reached the end of the file).

Note that switches at a fixed time use UTC time. They don't use the timecode of the input.

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.
Setting Up for Input Switching

- 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.

<table>
<thead>
<tr>
<th>Current Input</th>
<th>Next Input</th>
<th>Possible Start Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
<td>File</td>
<td>Fixed or Immediate</td>
</tr>
<tr>
<td>File</td>
<td>File</td>
<td>Follow</td>
</tr>
<tr>
<td>File</td>
<td>Live</td>
<td>Fixed or Immediate</td>
</tr>
<tr>
<td>File</td>
<td>Live</td>
<td>Follow</td>
</tr>
<tr>
<td>Live</td>
<td>File</td>
<td>Fixed or Immediate</td>
</tr>
<tr>
<td>Live</td>
<td>Live</td>
<td>Fixed or Immediate</td>
</tr>
</tbody>
</table>

Follow Chain

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 input above the first follow and ends with the last follow input. In the preceding example, the chain starts with input A and ends with input D.

The last input can be file or live. Either of these types can come before a fixed or immediate input (input E).

The other inputs (inputs A, B, C) in the follow chain must be files because they must have a defined ending so that the next input can successfully follow.

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 setting up a channel, as described in Setting up: Planning Your workflow (p. 57) the section called “Creating a channel from scratch” (p. 69).
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 the section called “Planning the channel” (p. 62).

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.

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 live sources. For a list of sources that are considered to be live inputs, see the section called “Supported Input Types and Upstream Systems” (p. 336).
   A multiple-input channel can have 0 to 2 live sources.
3. Note which sources are file sources (MP4 files), but don't worry about how many you have.
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 or Dolby Digital Plus 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 or Dolby Digital Plus 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 *MediaLive Input* (p. 126). 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. 254).

**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. 265).

5. Make sure you haven't exceeded the maximum number of inputs for the channel.

The channel can contain a maximum of 20 inputs. Subtract the live sources (inputs) you identified from 20 to obtain the maximum number of files inputs for this channel. Look at your sets of file source—each set becomes an input. Make sure you don't exceed the maximum for the channel.

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 (p. 272)
• Plan the Video Selectors  (p. 273)
• Plan the Audio Selectors  (p. 273)
• Planning the Captions Selectors  (p. 274)

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 EAC2 passthrough (EAC2 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 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 the selectors names that include 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

After you follow this step, you have identified one input as being the first 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. Each switch specifies the input it uses, and the switch type (fixed, immediate, or follow.)
• Names for all the switch actions.

Topics

• Plan the Action Names (p. 275)
• Plan the Order of Input Switches (p. 276)
• Handling the Transition When the Next Input Is Fixed or Immediate (p. 276)
• Handling the Transition When the Next Input Is Follow (p. 277)

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
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. Put the remaining input attachments in the order that you want. Decide on the type of switch for each item: fixed, immediate, or follow. For more information, see the section called “Types of Switches” (p. 265) and the section called “Rules and Limits” (p. 264).
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.

Note the following:

• 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, note the URL for each usage.

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 might let the order of the input attachments in the channel control the ingest of the first few inputs. You set up the schedule to perform its input switch only after those first ingests.

We don't recommend this model because you can't monitor the order of ingest from one place (the schedule). You must look at the order of input attachments and at the schedule.

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 that starts at a fixed time (input B) 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.

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 (p. 277)
- Identify the First Input for the Channel (p. 278)
- Create the Channel (p. 278)

Create the Inputs

This section is a supplement to the information in MediaLive Input (p. 126). 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. 127) 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. 254).
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. 69). 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. 69).

Channel and Input Details Pane

On the Channel and input details pane for the channel, in the Input specifications (p. 73) 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. 277).

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. 272).
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. 276). 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. 276).
- 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. 272):
  - 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. 267).

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, see the section called “Creating Actions” (p. 163).

We recommend that you add input switch actions to the schedule before you start the channel. You should decide how far into the future the schedule will control the ingest of inputs. You should plan to update the schedule regularly.

Remember the following:

- You can add input switches to the schedule while the channel is running.
- You can't add inputs to the channel unless you stop the channel.

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:

- Live inputs (in other words, all push inputs) must be already pushing before you start the channel. A live 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 (p. 279)
- Restarting a Channel (p. 280)
- What Happens With an Empty Schedule (p. 280)

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, 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 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.
Procedure

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. 69).

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. 89).

Topics
- Procedure to set up custom paths (p. 281)
- How manifests work (p. 281)
- Rules for custom paths (p. 283)
- Guidance for setting up for custom paths (p. 283)
- Examples of custom paths (p. 284)

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. 281).

2. Design the paths, paying attention to the syntax (p. 94) and rules (p. 95).

   See this guidance for different downstream systems (p. 283).

   See these examples (p. 284).

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:
How manifests work

- The content inside each child manifest includes a path to its media files.

By default, the syntax of this path is the following:

**base_filename name_modifier extension**

For example:

```
curling_high.m3u8
```

The path is relative to the location of the main 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:

  **base_url_manifest base_filename name_modifier 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:

  **base_url_content base_filename name_modifier optional_segment_modifier 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. 281). 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 master 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 master 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:
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. 69).

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. 89).

### Topics
- Procedure to set up redundant manifests (p. 285)
- The media contents of an HLS manifest (p. 287)
- Rules for most downstream systems (p. 288)
- Rules for Akamai CDNs (p. 289)
- Combining redundant manifests with other features (p. 290)

### 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” (p. 89). 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. 288)
   - the section called “Rules for Akamai” (p. 289)

   These rules supplement the information in the section called “Destination fields” (p. 89).

4. If you also need custom paths for manifests, make sure you read the information in the section called “How custom paths work” (p. 282). 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. 287).

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. 102) is set to EMIT_OUTPUT, MediaLive continues to update the master and child manifests.

From the point of view of the downstream system, there is no change to the master 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:

```bash
#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 base_filename 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. 289).

The rules for downstream systems (except Akamai) are based on these requirements:

- MediaLive pushes the files from both pipelines to the same location (protocol/domain/path).
- Given that the location is the same, the base_filenames for the pipelines must be different.
- If you are also implementing custom manifest paths (p. 282), the URL inside the manifests must be identical.

Make sure that the downstream system can work with these restrictions.

<table>
<thead>
<tr>
<th>Field</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol/domain/path portion of the two destination URIs (A and B)</td>
<td>Must be identical in both fields.</td>
</tr>
<tr>
<td>Base_filename portion of the two destination URIs (A and B)</td>
<td>Must be different in each field. It cannot use variable identifiers (p. 343) that include the date or time.</td>
</tr>
<tr>
<td>Name_modifier for each output</td>
<td>There is only one instance of this field. Both pipelines use the same value.</td>
</tr>
</tbody>
</table>
Rules for Akamai

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. 288).

<table>
<thead>
<tr>
<th>Field</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol/domain/path portion of the two destination URIs (A and B)</td>
<td>Can be different from each other, or can be the same.</td>
</tr>
<tr>
<td>Base_filename portion of the two destination URIs (A and B)</td>
<td>Can be different from each other, or can be the same.</td>
</tr>
<tr>
<td>Name_modifier</td>
<td>There is only one instance of this field. Both pipelines use the same value.</td>
</tr>
<tr>
<td>Segment modifier</td>
<td>There is only one instance of this field. Both pipelines use the same value.</td>
</tr>
<tr>
<td>Base URL Manifest A and Base URL Manifest B</td>
<td>These fields apply only if you are also implementing custom manifest paths (p. 282). Complete both fields.</td>
</tr>
<tr>
<td>Base URL Content A and Base URL Content B</td>
<td>These fields apply only if you are also implementing custom manifest paths (p. 282). Complete both fields.</td>
</tr>
</tbody>
</table>
## 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. 283) and for redundant manifests for your downstream system—either an Akamai CDN (p. 289) or another downstream system (p. 288).

### 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. 217).

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 master 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",...
```

---

## 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 (p. 291)
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.) MediaLive creates and delivers the MPTS. For more details, see the section called "Starting, Pausing, or Stopping a Multiplex" (p. 295).

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. 348).

• 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. 149).

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. 127).
• 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. 149). 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. 293).

• 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. 293).

For more information about completing the channel fields, see the section called “Creating a channel from scratch” (p. 69).

**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.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>In <strong>Multiplex destination</strong>, the <strong>Multiplex program</strong> field</td>
<td>From the list, choose the multiplex program that this channel belongs to.</td>
</tr>
<tr>
<td>In <strong>Stream settings</strong>, for <strong>Video</strong></td>
<td>The output can contain one, and only one, video asset.</td>
</tr>
<tr>
<td>In <strong>Stream settings</strong>, for <strong>Audio</strong></td>
<td>The output can contain zero or more audio assets.</td>
</tr>
<tr>
<td>In <strong>Stream settings</strong>, for <strong>Captions</strong></td>
<td>The output can contain zero or more captions assets.</td>
</tr>
</tbody>
</table>
Restrictions in the Video

The following rules apply to the fields in the video.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Width</strong> and <strong>Height</strong> (resolution)</td>
<td>Set values for both the width and height. The width can be up to 1920 pixels. The height can be up to 1080.</td>
</tr>
<tr>
<td><strong>Codec settings</strong></td>
<td>Choose <strong>H.264</strong> (AVC) or <strong>H.265</strong> (HEVC).</td>
</tr>
<tr>
<td><strong>In Aspect Ratio, the PAR control field</strong></td>
<td>Set a value. This is required. Don't set up to follow the aspect ratio from the source.</td>
</tr>
<tr>
<td><strong>In Rate control, the Rate control mode field</strong></td>
<td>Choose <strong>Multiplex</strong>.</td>
</tr>
<tr>
<td><strong>In Rate control, the Buffer size field</strong></td>
<td>Keep blank.</td>
</tr>
<tr>
<td><strong>In Frame rate, the Framerate field</strong></td>
<td>Set a value. This is required. Don't set up to follow the frame rate from the source.</td>
</tr>
<tr>
<td><strong>In GOP structure</strong></td>
<td>For <strong>GOP size units</strong>, choose <strong>FRAMES</strong>. Then set <strong>GOP structure</strong> to 6 or greater.</td>
</tr>
<tr>
<td></td>
<td>Or for <strong>GOP size units</strong>, choose <strong>SECONDS</strong>. Then set <strong>GOP structure</strong> to 0.1 or greater.</td>
</tr>
<tr>
<td><strong>In Codec details, the Profile field</strong></td>
<td>If the codec is <strong>H.264</strong>, choose one of these profiles:</td>
</tr>
<tr>
<td></td>
<td>• <strong>BASELINE</strong></td>
</tr>
<tr>
<td></td>
<td>• <strong>HIGH</strong></td>
</tr>
<tr>
<td></td>
<td>• <strong>MAIN</strong></td>
</tr>
<tr>
<td></td>
<td>If the codec is <strong>H.265</strong>, choose one of these profiles:</td>
</tr>
<tr>
<td></td>
<td>• <strong>BASELINE</strong></td>
</tr>
<tr>
<td></td>
<td>• <strong>HIGH</strong></td>
</tr>
<tr>
<td></td>
<td>• <strong>HIGH_10BIT</strong></td>
</tr>
<tr>
<td></td>
<td>• <strong>MAIN</strong></td>
</tr>
</tbody>
</table>

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.
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 (p. 295)
- Starting the Multiplex (p. 296)
- Pausing Activity in the Multiplex (p. 296)
- Stopping Activity in the Multiplex (p. 296)

Summary of These Actions

The following table summarizes the start, stop and pause capabilities for the multiplex, program, and channel.

<table>
<thead>
<tr>
<th>Item</th>
<th>Action</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiplex</td>
<td>Start</td>
<td>You can start a multiplex and the channels in any order.</td>
</tr>
<tr>
<td></td>
<td>Stop</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Pause</td>
<td>You can't pause a multiplex.</td>
</tr>
<tr>
<td>Program</td>
<td>Any</td>
<td>You don't start or stop a program.</td>
</tr>
<tr>
<td>Channel</td>
<td>Start</td>
<td>You can start a channel that is used in a multiplex at any time, including before you have started the multiplex.</td>
</tr>
<tr>
<td></td>
<td>Stop</td>
<td>You can stop a channel without stopping the multiplex. You must stop a channel in order to edit it.</td>
</tr>
<tr>
<td></td>
<td>Pause</td>
<td>You can't pause a channel that is used in a multiplex.</td>
</tr>
</tbody>
</table>
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. 279).

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 that automatically creates an entitlement in AWS Elemental MediaConnect, in the account associated with the MediaLive that is creating the MPTS. You don't have to perform any steps to set up this MediaConnect entitlement.

The entitlement name includes the string “multiplex” and the multiplex ID, so that MediaConnect users can easily identify it.

You can work with the entitlement as you would with any entitlement, to distribute the MPTS content.

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**

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**

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. 198).

**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. 255) 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.

**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. 218).
- **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. 80).
- **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 “Channel class” (p. 243).
SCTE-35 Message Processing

**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.
The information in this SCTE-35 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. 69). It also assumes that you have started creating a channel, including associating an input with the channel.

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 RTP or HLS 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.
MediaLive doesn't support processing of ad avail decorations in input manifests. The ad avail decorations in input manifests are always 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 (p. 299)
- Getting Ready: Set the Ad Avail Mode (p. 305)
- Enabling Manifest Decoration in the Output (p. 306)
- Enabling Ad Avail Blanking in the Output (p. 309)
- Enabling Blackout in the Output (p. 312)
- Enabling SCTE-35 Passthrough or Removal in the Output (p. 315)
- Inserting SCTE-35 Messages Using the Schedule (p. 317)
- Sample Manifests - HLS (p. 317)

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.

<table>
<thead>
<tr>
<th>Input</th>
<th>Interpret SCTE-35 Messages in the Transport Stream</th>
<th>Interpret Ad Avail Information in the Input Manifest</th>
</tr>
</thead>
<tbody>
<tr>
<td>HLS</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>RTMP</td>
<td>No</td>
<td>Not applicable</td>
</tr>
<tr>
<td>RTP</td>
<td>Yes</td>
<td>Not applicable</td>
</tr>
</tbody>
</table>

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. 309) and the section called “Enabling Blackout” (p. 312).

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. 306).

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. 315).

Inserting SCTE-35 Messages Using the Schedule

You can insert SCTE-35 messages in TS outputs using the channel schedule (p. 161). 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. 317).

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 (p. 303)
• Frame Capture Output (p. 303)
• HLS Output (p. 303)
• MediaPackage Output (p. 304)
• Microsoft Smooth Output (p. 304)
• RTMP Output (p. 304)
• UDP Output (p. 305)

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 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.

Follow this procedure if you want to support one or more of the following features:

- Manifest decoration
- Ad avail blanking

If your processing does not involve at least one of these features, the ad avail mode is ignored.

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.

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. 307), when video is blanked (p. 310), and when video is blacked out (p. 314).

4. In Ad avail offset, set a value, if desired. For details about a field, 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. 311).
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 (p. 306)
- Enabling Decoration – Microsoft Smooth (p. 306)
- How SCTE-35 Events Are Handled in Manifests and Sparse Tracks (p. 307)

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 "Getting Ready: Set the Ad Avail Mode" (p. 305).
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. 317).
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 "Getting Ready: Set the Ad Avail Mode" (p. 305).
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**

<table>
<thead>
<tr>
<th>Type of Instruction</th>
<th>When Inserted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base64</td>
<td>Information about all SCTE-35 messages in the output is incorporated into the manifest; the entire SCTE-35 message is added in base64 format.</td>
</tr>
<tr>
<td>Cue-out, Cue-in</td>
<td>SCTE-35 messages that are ad avails result in the insertion of cue-out, cue-in instructions.</td>
</tr>
<tr>
<td>Blackout</td>
<td>Only applies to the SCTE-35 Enhanced ad marker style (for HLS output; see the section called “Enabling Decoration – HLS” (p. 306)). 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.</td>
</tr>
</tbody>
</table>

**Splice Insert Mode**

**Message Type ID: Splice Insert**

<table>
<thead>
<tr>
<th>Segmentation Type ID</th>
<th>Base64</th>
<th>Cue-out, Cue-in</th>
<th>Blackout</th>
</tr>
</thead>
<tbody>
<tr>
<td>No segmentation descriptor present</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Provider advertisement</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Distributor advertisement</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Placement opportunity</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Other: Programs, Chapters, Network, Unscheduled</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Message Type ID: Time Signal**

<table>
<thead>
<tr>
<th>Segmentation Type ID</th>
<th>Base64</th>
<th>Cue-out, Cue-in</th>
<th>Blackout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provider advertisement</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>
### Segmentation Type ID

<table>
<thead>
<tr>
<th>Segmentation Type ID</th>
<th>Base64</th>
<th>Cue-out, Cue-in</th>
<th>Blackout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributor advertisement</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Placement opportunity</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Other: Programs, Chapters, Network, Unscheduled</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Segmentation Type ID</th>
<th>Base64</th>
<th>Cue-out, Cue-in</th>
<th>Blackout</th>
</tr>
</thead>
<tbody>
<tr>
<td>No segmentation descriptor present</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Provider advertisement</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distributor advertisement</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Placement opportunity</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other: Programs, Chapters, Network, Unscheduled</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Message Type ID: Time Signal**

<table>
<thead>
<tr>
<th>Segmentation Type ID</th>
<th>Base64</th>
<th>Cue-out, Cue-in</th>
<th>Blackout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provider advertisement</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distributor advertisement</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Placement opportunity</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Other: Programs, Chapters, Network, Unscheduled</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

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. 305)).

A similar feature is blackout (p. 312).

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.

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. 307), when video is blanked (p. 310), and when video is blacked out (p. 314).
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. 310).
   - 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. 311).

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. 31).

### 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**

<table>
<thead>
<tr>
<th>Segmentation Type ID</th>
<th>Blanked</th>
</tr>
</thead>
<tbody>
<tr>
<td>No segmentation descriptor present</td>
<td>Yes</td>
</tr>
<tr>
<td>Provider advertisement</td>
<td>Yes</td>
</tr>
<tr>
<td>Distributor advertisement</td>
<td>Yes</td>
</tr>
<tr>
<td>Placement opportunity</td>
<td>Yes</td>
</tr>
<tr>
<td>Other: Programs, Chapters, Network, Unscheduled</td>
<td>No</td>
</tr>
</tbody>
</table>

**Message Type ID: Time Signal**

<table>
<thead>
<tr>
<th>Segmentation Type ID</th>
<th>Blanked</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provider advertisement</td>
<td>Yes</td>
</tr>
<tr>
<td>Distributor advertisement</td>
<td>Yes</td>
</tr>
<tr>
<td>Placement opportunity</td>
<td>Yes</td>
</tr>
<tr>
<td>Other: Programs, Chapters, Network, Unscheduled</td>
<td>No</td>
</tr>
</tbody>
</table>
Triggers in Timesignal APOS Mode

This section describes which message type/segmentation type combination is blanked by ad avail blanking when the Ad Avail mode is Timesignal with APOS mode.

Message Type ID: Splice Insert

<table>
<thead>
<tr>
<th>Segmentation Type ID</th>
<th>Blanked</th>
</tr>
</thead>
<tbody>
<tr>
<td>No segmentation descriptor present</td>
<td>No</td>
</tr>
<tr>
<td>Provider advertisement</td>
<td>No</td>
</tr>
<tr>
<td>Distributor advertisement</td>
<td>No</td>
</tr>
<tr>
<td>Placement opportunity</td>
<td>No</td>
</tr>
<tr>
<td>Other: Programs, Chapters, Network,</td>
<td>No</td>
</tr>
<tr>
<td>Unscheduled</td>
<td></td>
</tr>
</tbody>
</table>

Message Type ID: Time Signal

<table>
<thead>
<tr>
<th>Segmentation Type ID</th>
<th>Blanked</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provider advertisement</td>
<td>No</td>
</tr>
<tr>
<td>Distributor advertisement</td>
<td>No</td>
</tr>
<tr>
<td>Placement opportunity</td>
<td>Yes</td>
</tr>
<tr>
<td>Other: Programs, Chapters, Network,</td>
<td>No</td>
</tr>
<tr>
<td>Unscheduled</td>
<td></td>
</tr>
</tbody>
</table>

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 these flags. See the section called “Enabling Blanking” (p. 309). 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. 305)). For example, chapters and programs.

(A similar feature is described in the section called “Enabling Ad Avail Blanking” (p. 309).)

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. 307), when video is blanked (p. 310), and when video is blacked out (p. 314).
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. 314).
   - 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. 311).
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. 31).
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
Enabling Blackout

Systems Manager password for accessing the S3 bucket. See the section called "About the Feature for Creating Password Parameters" (p. 31).

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. 305) is not relevant. Blackout works the same with either mode.

The segmentation ID triggers blackout based on “events,” as shown in the following table.

<table>
<thead>
<tr>
<th>SCTE-35 Segmentation Type</th>
<th>Blacked out</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter Start</td>
<td>Start blacking out</td>
</tr>
<tr>
<td>Chapter End</td>
<td>End blacking out</td>
</tr>
<tr>
<td>Network Start</td>
<td><strong>End blacking out</strong></td>
</tr>
<tr>
<td>Network End</td>
<td><strong>Start blacking out</strong></td>
</tr>
<tr>
<td>Program Start</td>
<td>Start blacking out</td>
</tr>
<tr>
<td>Program End</td>
<td>End blacking out</td>
</tr>
<tr>
<td>Unscheduled Event Start</td>
<td>Start blacking out</td>
</tr>
<tr>
<td>Unscheduled Event End</td>
<td>End blacking out</td>
</tr>
</tbody>
</table>

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."

<table>
<thead>
<tr>
<th>SCTE-35 Segmentation Type</th>
<th>Strength</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network</td>
<td>1 (Strongest)</td>
</tr>
<tr>
<td>Unscheduled Event</td>
<td>2</td>
</tr>
<tr>
<td>Program</td>
<td>3</td>
</tr>
</tbody>
</table>
A blackout can be ended only by an event of equal or greater strength than the event that started it.

For example, if the blackout is started by a Program Start, it can be ended by a Network Start, an Unscheduled Event End or a Program End. It cannot be ended by a Chapter End. MediaLive ignores the "end blackout" instruction implied by the Chapter End.

**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. 309). 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. 161) 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 **MediaLive Schedule** (p. 159).

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. 301) and the section called “Supported Features by Output Type” (p. 302).

### 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

<table>
<thead>
<tr>
<th>Segment</th>
<th>Tag</th>
<th>Tag Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segment in which the ad avail starts.</td>
<td>1 CUE: DURATION tag</td>
<td>1</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Segment</th>
<th>Tag</th>
<th>Tag Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Each succeeding segment.</td>
<td>CUE-OUT</td>
<td>1</td>
</tr>
<tr>
<td>Segment in which the ad avail starts.</td>
<td>CUE-OUT-CONT</td>
<td>0-n</td>
</tr>
<tr>
<td>Segment in which ad avail ends.</td>
<td>CUE-IN</td>
<td>1</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Segment</th>
<th>Tag</th>
<th>Tag Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segment in which the ad avail starts.</td>
<td>OATCLS-SCTE35</td>
<td>1</td>
</tr>
<tr>
<td>Segment in which the ad avail starts.</td>
<td>ASSET</td>
<td>1</td>
</tr>
<tr>
<td>Segment in which the ad avail starts.</td>
<td>CUE-OUT</td>
<td>1</td>
</tr>
<tr>
<td>Each succeeding segment.</td>
<td>CUE-OUT-CONT</td>
<td>0-n</td>
</tr>
<tr>
<td>Segment in which ad avail ends.</td>
<td>CUE-IN</td>
<td>1</td>
</tr>
</tbody>
</table>

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:/DA0AAAAAAAAAAAAABQb+ADAQ6QAbADAhxDVUVJQAAA0/PAAEurBoIAAAg+2UBNAANvrtoQ==
#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+2UBNAANvrtoQ==
  .
  .
#EXT-X-CUE-IN
```

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

For information about adding and managing tags, see Managing Tags (p. 320).

**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. 71) – for information about including tags when you create a channel
- the section called “Editing and Deleting a Channel” (p. 121) – for information about modifying tags in an existing channel
- MediaLive Input (p. 126) – for information about including tags in an input
- MediaLive Input Security Groups (p. 145) – for information about including tags in an input security group
- the section called “Creating a Multiplex and Program” (p. 149) – 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. 323).

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. 69).

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:
AWS Elemental MediaLive User Guide
About the Synchronization Threshold

• 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. 322).

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 (p. 322)
• How Timecode Works at Runtime (p. 323)

About the Synchronization Threshold

The **Sync thresholds** 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. 262).

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.

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 (p. 116) as part of creating a channel. This feature lets you control the quality and bitrate of the video.

When encoding visually complex video (such as high-motion sports events with brightly dressed crowds in the background), there is always a trade-off between high video quality and low bitrate. Higher video quality requires higher bitrate. There is less trade-off with visually simple video such as cartoons.

AWS Elemental MediaLive offers several options that provide different balances of video quality versus bitrate.

To set the rate control mode and bitrate for the output

1. On the Stream settings pane, for Video, for Codec settings, choose H264.
2. In the Rate Control section, for Rate control mode, choose QVBR or CBR or VBR. For information about choosing the best option, see the sections below.

   - If you chose QVBR, complete Max bitrate and Quality level.
   - If you chose VBR, complete Bitrate (average bitrate) and Max bitrate.
   - If you chose CBR, complete Bitrate.
Quality-defined Variable Bitrate Mode (QVBR)

With quality-defined variable bitrate mode (QVBR), you specify a maximum bitrate and a quality level. Video quality will match the specified quality level except when it is constrained by the maximum bitrate; this constraint occurs when the video is very complex so that it is not possible to reach the quality level without exceeding the maximum bitrate.

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.

**Values to use:** When choosing QVBR, you should set the quality level and maximum bitrate for your most important viewing devices. Set the buffer size to twice the maximum bitrate, and set the initial buffer to 90%.

<table>
<thead>
<tr>
<th>Viewing Device</th>
<th>Quality Level</th>
<th>Max Bitrate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Screen</td>
<td>8 to 10</td>
<td>4,000,000 to 6,000,000</td>
</tr>
<tr>
<td>PC or Tablet</td>
<td>7</td>
<td>1,500,000 to 3,000,000</td>
</tr>
<tr>
<td>Smartphone</td>
<td>6</td>
<td>1,000,000 to 1,500,000</td>
</tr>
</tbody>
</table>

**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.

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.

**Values to use:** When choosing VBR, you should try to assess the expected complexity of the video, and set a suitable average bitrate. Set the maximum bitrate to accommodate expected spikes. Set the buffer size to twice the maximum bitrate, and set the initial buffer to 90%.

**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.
Values to use: When choosing CBR, set the bitrate to balance the video quality and the output bitrate. Set the buffer size to twice the bitrate, and set the initial buffer to 90%.

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.

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

- How to Read the Supported Captions Information (p. 325)
- General Information About Supported Formats (p. 325)
- Formats Supported in an Archive Output (p. 329)
- Formats Supported in an HLS Output or a MediaPackage Output (p. 331)
- Formats Supported in a Microsoft Smooth Output (p. 332)
- Formats Supported in an RTMP Output (p. 333)
- Formats Supported in an MPEG2-UDP Streaming Output or a Multiplex Output (p. 334)

How to Read the Supported Captions Information

To determine if the input container and input captions that you have received are capable of producing the desired output captions, consult the tables in the following sections and follow these steps:

1. Find the table for your output container.
2. In that table, find the container type of the input that you have been provided with, and then find the input captions that are in that container.
3. In the third column, look for the output caption format that you require.

   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 an input container that includes input captions that can be converted to the required output format.

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.
<table>
<thead>
<tr>
<th>Caption</th>
<th>Supported in Input</th>
<th>Supported in Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ancillary data</td>
<td>Yes</td>
<td></td>
<td>From MXF input, data that is compliant with “SMPTE 291M: Ancillary Data Package and Space Formatting” and that is contained in ancillary data.</td>
</tr>
<tr>
<td>ARIB</td>
<td>Yes</td>
<td>Yes</td>
<td>Captions that are compliant with ARIB STD-B37 Version 2.4.</td>
</tr>
<tr>
<td>Burn-in</td>
<td></td>
<td>Yes</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>DVB-Sub</td>
<td>Yes</td>
<td>Yes</td>
<td>Captions that are compliant with ETSI EN 300 743.</td>
</tr>
<tr>
<td>Embedded</td>
<td>Yes</td>
<td>Yes</td>
<td>Captions that are compliant with the EIA-608 standard (also known as CEA-608 or SMPTE-259M or “line 21 captions”) or the CEA-708 standard (also known as EIA-708).</td>
</tr>
<tr>
<td>Embedded+SCTE-20</td>
<td>Yes</td>
<td>Yes</td>
<td>Captions that have both embedded and SCTE-20 in the video. The embedded captions are inserted before the SCTE-20 captions.</td>
</tr>
<tr>
<td>RTMP CaptionInfo</td>
<td></td>
<td>Yes</td>
<td>Captions that are compliant with the Adobe onCaptionInfo format.</td>
</tr>
<tr>
<td>SCTE-20</td>
<td>Yes</td>
<td></td>
<td>Captions that are compliant with the</td>
</tr>
</tbody>
</table>
### Caption "SCTE 20 2012 Methods for Carriage of CEA-608 Closed Captions and Non-Real Time Sampled Video.”

<table>
<thead>
<tr>
<th>Caption</th>
<th>Supported in Input</th>
<th>Supported in Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCTE-20+Embedded</td>
<td>Yes</td>
<td>Yes</td>
<td>Captions that are compliant with SCTE-43. The SCTE-20 captions are inserted in the video before the embedded captions.</td>
</tr>
<tr>
<td>SCTE-27</td>
<td>Yes</td>
<td></td>
<td>Captions that are compliant with the standard “SCTE-27 (2011), Subtitling Methods for Broadcast Cable.”</td>
</tr>
<tr>
<td>SMPTE-TT</td>
<td></td>
<td>Yes</td>
<td>Captions that are compliant with the standard &quot;SMPTE ST 2052-1:2010”</td>
</tr>
<tr>
<td>Teletext</td>
<td>Yes</td>
<td>Yes</td>
<td>From TS input: Captions in the EBU Teletext format.</td>
</tr>
<tr>
<td>TTML</td>
<td></td>
<td>Yes</td>
<td>Captions files that are compliant with the standard “Timed Text Markup Language 1 (TTML1) (Second Edition).”</td>
</tr>
<tr>
<td>WebVTT</td>
<td></td>
<td>Yes</td>
<td>Captions that are compliant with &quot;webvtt: The Web Video Text Tracks Format&quot; (<a href="http://dev.w3.org/html5/webvtt/">http://dev.w3.org/html5/webvtt/</a>).</td>
</tr>
</tbody>
</table>

### Captions Categories

Captions are grouped into five categories, based on how the captions are included in the output.

<table>
<thead>
<tr>
<th>Captions Format</th>
<th>Category of This Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARIB</td>
<td>Object-style</td>
</tr>
<tr>
<td>Burn-in</td>
<td>Burn-in</td>
</tr>
<tr>
<td>DVB-Sub</td>
<td>Object-style</td>
</tr>
</tbody>
</table>
### General Information About Supported Formats

<table>
<thead>
<tr>
<th>Captions Format</th>
<th>Category of This Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedded</td>
<td>Embedded</td>
</tr>
<tr>
<td>Embedded+SCTE-20</td>
<td>Embedded</td>
</tr>
<tr>
<td>RTMP CaptionInfo</td>
<td>Object-style</td>
</tr>
<tr>
<td>SCTE-20+Embedded</td>
<td>Embedded</td>
</tr>
<tr>
<td>SCTE-27</td>
<td>Object-style</td>
</tr>
<tr>
<td>SMPTE-TT</td>
<td>Stream</td>
</tr>
<tr>
<td>Teletext</td>
<td>Object-style</td>
</tr>
<tr>
<td>TTML</td>
<td>Sidecar</td>
</tr>
<tr>
<td>Web-VTT</td>
<td>Sidecar</td>
</tr>
</tbody>
</table>

**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.

**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 an Archive (for MPEG2-TS) output, when you have this input container and captions type.

<table>
<thead>
<tr>
<th>Source Caption Container</th>
<th>Source Caption Input</th>
<th>Supported Output Captions</th>
</tr>
</thead>
<tbody>
<tr>
<td>HLS Container</td>
<td>Embedded</td>
<td>Burn-in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DVB-Sub</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Embedded</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Embedded+SCTE-20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCTE-20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCTE-20+Embedded</td>
</tr>
<tr>
<td>SCTE-20</td>
<td></td>
<td>Burn-in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DVB-Sub</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Embedded</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Embedded+SCTE-20</td>
</tr>
</tbody>
</table>
## Formats Supported in an Archive Output

<table>
<thead>
<tr>
<th>Source Caption Container</th>
<th>Source Caption Input</th>
<th>Supported Output Captions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SCTE-20</td>
<td>Embedded</td>
</tr>
<tr>
<td></td>
<td>SCTE-20+Embedded</td>
<td></td>
</tr>
<tr>
<td>MP4 Container</td>
<td>Ancillary</td>
<td>Burn-in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DVB-Sub</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Embedded</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Embedded+SCTE-20</td>
</tr>
<tr>
<td></td>
<td>Embedded or Embedded +SCTE-20</td>
<td>Burn-in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DVB-Sub</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Embedded</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Embedded+SCTE-20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCTE-20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCTE-20+Embedded</td>
</tr>
<tr>
<td>RTMP Container</td>
<td>Embedded</td>
<td>Burn-in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DVB-Sub</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Embedded</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Embedded+SCTE-20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCTE-20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCTE-20+Embedded</td>
</tr>
<tr>
<td>MPEG2-TS Container</td>
<td>ARIB</td>
<td>ARIB</td>
</tr>
<tr>
<td>(through the RTP or MediaConnect protocol)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DVB-Sub</td>
<td>Burn-in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DVB-Sub</td>
</tr>
<tr>
<td></td>
<td>Embedded or Embedded +SCTE-20</td>
<td>Burn-in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DVB-Sub</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Embedded</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Embedded+SCTE-20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCTE-20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCTE-20+Embedded</td>
</tr>
</tbody>
</table>
### 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.

<table>
<thead>
<tr>
<th>Source Caption Container</th>
<th>Source Caption Input</th>
<th>Supported Output Captions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedded</td>
<td>Burn-in</td>
<td>SCTE-20</td>
</tr>
<tr>
<td>Embedded</td>
<td>DVB-Sub</td>
<td></td>
</tr>
<tr>
<td>Embedded</td>
<td>Embedded</td>
<td></td>
</tr>
<tr>
<td>Embedded</td>
<td>Embedded+SCTE-20</td>
<td></td>
</tr>
<tr>
<td>Embedded</td>
<td>SCTE-20</td>
<td></td>
</tr>
<tr>
<td>Embedded</td>
<td>SCTE-20+Embedded</td>
<td></td>
</tr>
<tr>
<td>None</td>
<td>Burn-in</td>
<td>SCTE-20</td>
</tr>
<tr>
<td>DVB-Sub</td>
<td>Embedded</td>
<td></td>
</tr>
<tr>
<td>DVB-Sub</td>
<td>Teletext</td>
<td></td>
</tr>
<tr>
<td>Teletext</td>
<td>DVB-Sub</td>
<td></td>
</tr>
<tr>
<td>Teletext</td>
<td>Teletext</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source Caption Container</th>
<th>Source Caption Input</th>
<th>Supported Output Captions</th>
</tr>
</thead>
<tbody>
<tr>
<td>HLS Container</td>
<td>Embedded</td>
<td>Burn-in</td>
</tr>
<tr>
<td></td>
<td>Embedded</td>
<td>Embedded</td>
</tr>
<tr>
<td></td>
<td>Embedded</td>
<td>WebVTT</td>
</tr>
<tr>
<td></td>
<td>SCTE-20</td>
<td>Burn-in</td>
</tr>
<tr>
<td></td>
<td>Embedded</td>
<td>Embedded</td>
</tr>
<tr>
<td></td>
<td>WebVTT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ancillary</td>
<td>Burn-in</td>
</tr>
<tr>
<td></td>
<td>embedded</td>
<td>Embedded</td>
</tr>
<tr>
<td></td>
<td>WebVTT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Embedded or Embedded</td>
<td>Burn-in</td>
</tr>
<tr>
<td></td>
<td>+SCTE-20</td>
<td>Embedded</td>
</tr>
<tr>
<td></td>
<td>WebVTT</td>
<td></td>
</tr>
<tr>
<td>MP4 Container</td>
<td>Ancillary</td>
<td>Burn-in</td>
</tr>
<tr>
<td></td>
<td>Embedded</td>
<td>Embedded</td>
</tr>
<tr>
<td></td>
<td>WebVTT</td>
<td></td>
</tr>
<tr>
<td>RTMP Container</td>
<td>Embedded</td>
<td>Burn-in</td>
</tr>
<tr>
<td></td>
<td>Embedded</td>
<td>Embedded</td>
</tr>
<tr>
<td></td>
<td>WebVTT</td>
<td></td>
</tr>
</tbody>
</table>
### 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.

<table>
<thead>
<tr>
<th>Source Caption Container</th>
<th>Source Caption Input</th>
<th>Supported Output Captions</th>
</tr>
</thead>
<tbody>
<tr>
<td>HLS Container</td>
<td>Embedded</td>
<td>Burn-in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMPTE-TT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TTML</td>
</tr>
<tr>
<td></td>
<td>SCTE-20</td>
<td>Burn-in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMPTE-TT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TTML</td>
</tr>
<tr>
<td>MP4 Container</td>
<td>Ancillary</td>
<td>Burn-in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMPTE-TT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TTML</td>
</tr>
<tr>
<td></td>
<td>Embedded or Embedded +SCTE-20</td>
<td>Burn-in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMPTE-TT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TTML</td>
</tr>
<tr>
<td>RTMP Container</td>
<td>Embedded</td>
<td>Burn-in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMPTE-TT</td>
</tr>
</tbody>
</table>

**MPEG2-TS Container (through the RTP or MediaConnect protocol)**

- ARIB: None
- DVB-Sub: Burn-in
- Embedded or Embedded +SCTE-20: Burn-in, Embedded, WebVTT
- SCTE-20: Burn-in, Embedded, WebVTT
- SCTE-27: Burn-in
- Teletext: Burn-in, WebVTT
### 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.

<table>
<thead>
<tr>
<th>Source Caption Container</th>
<th>Source Caption Input</th>
<th>Supported Output Captions</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPEG2-TS Container (through the RTP or MediaConnect protocol)</td>
<td>ARIB</td>
<td>None</td>
</tr>
<tr>
<td>DVB-Sub</td>
<td>Embedded or Embedded +SCTE-20</td>
<td>Burn-in, SMPTE-TT, TTML</td>
</tr>
<tr>
<td>Source Caption Container</td>
<td>Source Caption Input</td>
<td>Supported Output Captions</td>
</tr>
<tr>
<td>Embedded</td>
<td>Burn-in, SMPTE-TT, TTML</td>
<td></td>
</tr>
<tr>
<td>Embedded or Embedded +SCTE-20</td>
<td>Burn-in, SMPTE-TT, TTML</td>
<td></td>
</tr>
<tr>
<td>Modeling</td>
<td>Embedded or Embedded +SCTE-20</td>
<td>Burn-in, SMPTE-TT, TTML</td>
</tr>
<tr>
<td>Sources</td>
<td>Embedded or Embedded +SCTE-20</td>
<td>Burn-in, SMPTE-TT, TTML</td>
</tr>
<tr>
<td>Programming</td>
<td>Embedded or Embedded +SCTE-20</td>
<td>Burn-in, SMPTE-TT, TTML</td>
</tr>
<tr>
<td>Teletext</td>
<td>Embedded or Embedded +SCTE-20</td>
<td>Burn-in, SMPTE-TT, TTML</td>
</tr>
</tbody>
</table>
### Formats Supported in an MPEG2-UDP Streaming 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 MPEG2-UDP output or a multiplex output, when you have this input container and captions type.

<table>
<thead>
<tr>
<th>Source Caption Container</th>
<th>Source Caption Input</th>
<th>Supported Output Captions</th>
</tr>
</thead>
<tbody>
<tr>
<td>HLS Container</td>
<td>Embedded</td>
<td>Burn-in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DVB-Sub</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Embedded</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Embedded+SCTE-20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCTE-20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCTE-20+SCTE-20</td>
</tr>
<tr>
<td>MP4 Container</td>
<td>Ancillary</td>
<td>Burn-in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DVB-Sub</td>
</tr>
<tr>
<td>MPEG2-TS Container (through the RTP or MediaConnect protocol)</td>
<td>ARIB</td>
<td>None</td>
</tr>
<tr>
<td>DVB-Sub</td>
<td>Burn-in</td>
<td></td>
</tr>
<tr>
<td>Embedded or Embedded +SCTE-20</td>
<td>Burn-in</td>
<td>Embedded</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RTMP CaptionInfo</td>
</tr>
<tr>
<td>SCTE-20</td>
<td>Embedded</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RTMP CaptionInfo</td>
</tr>
<tr>
<td>SCTE-27</td>
<td>Burn-in</td>
<td></td>
</tr>
<tr>
<td>Teletext</td>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>
## Formats Supported in an MPEG2-UDP Streaming Output or a Multiplex Output

<table>
<thead>
<tr>
<th>Source Caption Container</th>
<th>Source Caption Input</th>
<th>Supported Output Captions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Embedded</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Embedded+SCTE-20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCTE-20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCTE-20+Embedded</td>
</tr>
<tr>
<td>Embedded or Embedded +SCTE-20</td>
<td></td>
<td>Burn-in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DVB-Sub</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Embedded</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Embedded+SCTE-20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCTE-20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCTE-20+Embedded</td>
</tr>
<tr>
<td>RTMP Container</td>
<td>Embedded</td>
<td>Burn-in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DVB-Sub</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Embedded</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Embedded+SCTE-20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCTE-20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCTE-20+Embedded</td>
</tr>
<tr>
<td>MPEG2-TS Container (through the RTP or MediaConnect protocol)</td>
<td>ARIB</td>
<td>ARIB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DVB-Sub</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Burn-in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DVB-Sub</td>
</tr>
<tr>
<td></td>
<td>Embedded or Embedded</td>
<td>Burn-in</td>
</tr>
<tr>
<td></td>
<td>+SCTE-20</td>
<td>DVB-Sub</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Embedded</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Embedded+SCTE-20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCTE-20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCTE-20+Embedded</td>
</tr>
</tbody>
</table>
Reference: Supported Containers and Codecs

This chapter provides information about the input content types and input video and audio codecs that AWS Elemental MediaLive can ingest, and the output content types and output video and codecs that MediaLive can produce.

Topics

- Supported Input Types and Upstream Systems (p. 336)
- Supported Codecs for Inputs (p. 339)
- Supported Containers and Downstream Systems (p. 339)
- Supported Codecs for Outputs (p. 341)

Supported Input Types and Upstream Systems

The following table lists the input types and protocols that MediaLive supports. Descriptions of the push and pull terms follow this table.

<table>
<thead>
<tr>
<th>MediaLive Input Type</th>
<th>Use Case</th>
<th>Upstream System and Supported Protocol</th>
<th>Live Stream Supported?</th>
<th>VOD Asset Supported?</th>
</tr>
</thead>
<tbody>
<tr>
<td>HLS</td>
<td>Pull an HLS stream or asset from an external endpoint using the HTTP protocol, with or without a secure connection.</td>
<td>HTTP server or HTTPS server</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>HLS</td>
<td>See the note after this table.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HLS</td>
<td>Pull an HLS stream or file from an AWS Elemental MediaStore</td>
<td>AWS Elemental MediaStore with a custom protocol</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MediaLive Input Type</td>
<td>Use Case</td>
<td>Upstream System and Supported Protocol</td>
<td>Live Stream Supported?</td>
<td>VOD Asset Supported?</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------</td>
<td>----------------------------------------</td>
<td>-----------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>HLS</td>
<td>Pull an HLS stream or file from an Amazon S3 bucket, with or without a secure connection.</td>
<td>Amazon S3 over a custom protocol</td>
<td>Yes See the note after this table.</td>
<td>Yes</td>
</tr>
<tr>
<td>MediaConnect</td>
<td>Push a transport stream (TS) from a flow in AWS Elemental MediaConnect. This input uses a MediaConnect flow ARN, not a URI.</td>
<td>AWS Elemental MediaConnect over an internal connection</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>MP4</td>
<td>Pull an MP4 file from an HTTP server, with or without a secure connection.</td>
<td>HTTP server or HTTPS server</td>
<td>No</td>
<td>Yes, with .mp4 file extension only</td>
</tr>
<tr>
<td>MP4</td>
<td>Pull an MP4 file from an Amazon Simple Storage Service bucket, with or without a secure connection. See Amazon S3 Bucket Names following this table.</td>
<td>Amazon S3 over a custom protocol</td>
<td>No</td>
<td>Yes, with .mp4 file extension only</td>
</tr>
<tr>
<td>RTP</td>
<td>Push a transport stream (TS) to a fixed endpoint on MediaLive, using the RTP protocol.</td>
<td>RTP server over RTP Push</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>RTP</td>
<td>Push a transport stream (TS) in your VPC to a fixed endpoint on MediaLive, using the RTP protocol.</td>
<td>Amazon VPC over RTP within a private cloud</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
## Supported Input Types and Upstream Systems

<table>
<thead>
<tr>
<th>MediaLive Input Type</th>
<th>Use Case</th>
<th>Upstream System and Supported Protocol</th>
<th>Live Stream Supported?</th>
<th>VOD Asset Supported?</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTMP Pull</td>
<td>Pull a stream from an external endpoint using the RTMP protocol.</td>
<td>RTMP server over RTMP Pull</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>RTMP Push</td>
<td>Push a stream to a fixed endpoint on MediaLive using the RTMP protocol.</td>
<td>RTMP server over RTMP Push</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>RTMP Push</td>
<td>Push a stream in your VPC to a fixed endpoint on MediaLive, using the RTMP protocol.</td>
<td>Amazon VPC over RTMP within a private cloud</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

### HLS Inputs – Live versus file sources

MediaLive considers an HLS input to be a live stream if the Buffer segments field has a value from 3 to 10, inclusive.

MediaLive considers the input to be a file asset if that field has a value of 11 or more, or 0, or undefined.

(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.)

### HLS Inputs – HLS Live Input from Amazon S3

We don’t recommend Amazon S3 as a source for a live stream.

### Inputs with Amazon S3 as the Upstream System – Bucket Names

With MediaLive, the bucket name can’t use dot notation. For example, mycompany-videos is valid, but mycompany.videos isn’t.

### RTMP Inputs – Protocol

MediaLive doesn’t support inputs using the RTMPS protocol.

### Ingesting with Push

A push input works as follows: the source attempts to deliver to an endpoint that is specified in the MediaLive input. For RTP protocols, the source is unaware of whether the content is being ingested by the MediaLive channel. For RTMP, 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 reacts to the source (RTP) or responds to the handshake message (RTMP) and ingests it. When the channel is not running, MediaLive does not react; the source continues to publish to the endpoint (RTP) or goes into a paused state (RTMP), but MediaLive ignores those actions.

A push input works only with a streaming source.
Ingesting with Pull

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 file input (where the source is made available on the endpoint and then does not change).

Supported Codecs for Inputs

The following table lists the codecs that MediaLive supports for source content.

<table>
<thead>
<tr>
<th>Container</th>
<th>Video Codecs</th>
<th>Audio Codecs</th>
</tr>
</thead>
<tbody>
<tr>
<td>HLS</td>
<td>H.264 (AVC)</td>
<td>AAC</td>
</tr>
<tr>
<td></td>
<td>See the note after this table.</td>
<td>Dolby Digital</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dolby Digital Plus</td>
</tr>
<tr>
<td>MediaConnect</td>
<td>H.264 (AVC)</td>
<td>AAC</td>
</tr>
<tr>
<td></td>
<td>HEVC (H.265)</td>
<td>Dolby Digital</td>
</tr>
<tr>
<td></td>
<td>MPEG-2</td>
<td>Dolby Digital Plus</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPEG Audio</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PCM</td>
</tr>
<tr>
<td>MP4</td>
<td>H.264 (AVC)</td>
<td>AAC</td>
</tr>
<tr>
<td></td>
<td>MPEG-2</td>
<td></td>
</tr>
<tr>
<td>RTMP</td>
<td>H.264 (AVC)</td>
<td>AAC</td>
</tr>
<tr>
<td>RTP (MPEG TS)</td>
<td>H.264 (AVC)</td>
<td>AAC</td>
</tr>
<tr>
<td></td>
<td>HEVC (H.265)</td>
<td>Dolby Digital</td>
</tr>
<tr>
<td></td>
<td>MPEG-2</td>
<td>Dolby Digital Plus</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPEG Audio</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PCM</td>
</tr>
</tbody>
</table>

HLS Inputs

The audio and video assets must be multiplexed into a single stream. MediaLive can't ingest audio assets that are in an audio rendition group.

Supported Containers and Downstream Systems

The following table lists the output formats and protocols that MediaLive supports.
### Supported Containers and Downstream Systems

<table>
<thead>
<tr>
<th>MediaLive Output Type (Output Group)</th>
<th>Use Case</th>
<th>Downstream System and Supported Protocol</th>
<th>Live Output Supported</th>
<th>VOD Output Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archive</td>
<td>Send transport stream (TS) files to an Amazon S3 bucket. See Amazon S3 Bucket Names following this table.</td>
<td>Amazon S3, over a custom protocol</td>
<td>No</td>
<td>Yes. A channel can contain only one archive output group.</td>
</tr>
<tr>
<td>Frame Capture</td>
<td>Send a series of JPEG files to an Amazon S3 bucket. See Amazon S3 Bucket Names following this table.</td>
<td>Amazon S3, over a custom protocol</td>
<td>No</td>
<td>Yes. A channel can contain a maximum of three frame capture output groups.</td>
</tr>
<tr>
<td>HLS</td>
<td>Send an HLS stream to a server that supports HTTP PUT or WebDav.</td>
<td>HTTP server</td>
<td>Yes</td>
<td>Yes, when the output group is set up for VOD mode</td>
</tr>
<tr>
<td>HLS</td>
<td>Send an HLS stream to a server that supports HTTPS PUT or WebDav.</td>
<td>HTTPS server</td>
<td>Yes</td>
<td>Yes, when the output group is set up for VOD mode</td>
</tr>
<tr>
<td>HLS</td>
<td>Send an HLS stream to an Akamai CDN.</td>
<td>Akamai CDN, over HTTP or HTTPS</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>HLS</td>
<td>Send an HLS stream to a MediaPackage channel using the HTTPS protocol.</td>
<td>AWS Elemental MediaPackage, over HTTPS</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>HLS</td>
<td>Send an HLS stream to a container on MediaStore.</td>
<td>AWS Elemental MediaStore, with a custom protocol</td>
<td>Yes</td>
<td>Yes, when the output group is set up for VOD mode</td>
</tr>
<tr>
<td>MediaPackage</td>
<td>Send an HLS stream to a MediaPackage channel.</td>
<td>AWS Elemental MediaPackage over an HTTPS WebDav</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Microsoft Smooth</td>
<td>Send a stream to an origin server or CDN that supports Microsoft Smooth Streaming.</td>
<td>A supported CDN, over HTTP or HTTPS</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
## Supported Codecs for Outputs

### MediaLive Output Type (Output Group)

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Downstream System and Supported Protocol</th>
<th>Live Output Supported</th>
<th>VOD Output Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiplex</td>
<td>Create a transport stream (TS) that is part of a MediaLive multiplex.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>RTMP</td>
<td>Send a stream to a server that supports the RTMP protocol.</td>
<td>RTMP server</td>
<td>Yes</td>
</tr>
<tr>
<td>RTMPS</td>
<td>Send a stream to a server that supports the RTMPS protocol.</td>
<td>RTMPS server</td>
<td>Yes</td>
</tr>
<tr>
<td>UDP</td>
<td>Send a transport stream (TS) to a server that supports UDP.</td>
<td>UDP server</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### 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.

## Supported Codecs for Outputs

### Supported Codecs for File Outputs

The following table lists the codecs for file outputs that MediaLive supports.

<table>
<thead>
<tr>
<th>Container</th>
<th>Video Codecs</th>
<th>Audio Codecs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archive</td>
<td>H.264 (AVC)</td>
<td>AAC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dolby Digital</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dolby Digital Plus</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPEG-1 Layer II</td>
</tr>
<tr>
<td>Frame Capture</td>
<td>JPEG</td>
<td>None. A Frame Capture output doesn't include audio.</td>
</tr>
<tr>
<td>HLS</td>
<td>H.264 (AVC)</td>
<td>AAC</td>
</tr>
<tr>
<td></td>
<td>H.265 (HEVC)</td>
<td>Dolby Digital</td>
</tr>
<tr>
<td>Microsoft Smooth</td>
<td>H.264 (AVC)</td>
<td>AAC</td>
</tr>
</tbody>
</table>
Supported Codecs for Outputs

The following table lists the codecs that MediaLive supports within streaming outputs.

<table>
<thead>
<tr>
<th>Container</th>
<th>Video Codecs</th>
<th>Audio Codecs</th>
</tr>
</thead>
<tbody>
<tr>
<td>HLS</td>
<td>H.264 (AVC)</td>
<td>AAC</td>
</tr>
<tr>
<td></td>
<td>H.265 (HEVC)</td>
<td>Dolby Digital</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dolby Digital Plus</td>
</tr>
<tr>
<td>MediaPackage</td>
<td>H.264 (AVC)</td>
<td>AAC</td>
</tr>
<tr>
<td></td>
<td>H.265 (HEVC)</td>
<td>Dolby Digital</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dolby Digital Plus</td>
</tr>
<tr>
<td>Microsoft Smooth</td>
<td>H.264 (AVC)</td>
<td>AAC</td>
</tr>
<tr>
<td></td>
<td>H.265 (HEVC)</td>
<td>Dolby Digital</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dolby Digital Plus</td>
</tr>
<tr>
<td>Multiplex</td>
<td>H.264 (AVC)</td>
<td>AAC</td>
</tr>
<tr>
<td></td>
<td>H.265 (HEVC)</td>
<td>Dolby Digital</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dolby Digital Plus</td>
</tr>
<tr>
<td>RTMP or RTMPS</td>
<td>H.264 (AVC)</td>
<td>AAC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dolby Digital</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dolby Digital Plus</td>
</tr>
<tr>
<td>UDP</td>
<td>H.264 (AVC)</td>
<td>AAC</td>
</tr>
<tr>
<td></td>
<td>H.265 (HEVC)</td>
<td>Dolby Digital</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dolby Digital Plus</td>
</tr>
</tbody>
</table>
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, $\text{dt}$ for the date and time) when you run the channel. For example, $\text{dt}$ 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.

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{dt}$</td>
<td>YYYYMMDDTHHMMSS</td>
<td>For HLS outputs, the UTC date and time of each segment.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For all other outputs, the UTC date and start time of the channel.</td>
</tr>
<tr>
<td>$\text{d}$</td>
<td>YYYYMMDD</td>
<td>For HLS outputs, the UTC date and time of each segment.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For all other outputs, the UTC date when the channel starts.</td>
</tr>
<tr>
<td>$\text{t}$</td>
<td>HHMMSS</td>
<td>For HLS outputs, the UTC time of each segment.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For all other outputs, the UTC start time of the channel.</td>
</tr>
<tr>
<td>$\text{rv}$</td>
<td>Kb</td>
<td>Video bitrate.</td>
</tr>
<tr>
<td>$\text{ra}$</td>
<td>Kb</td>
<td>Total of all audio bitrates in the output.</td>
</tr>
<tr>
<td>$\text{rc}$</td>
<td>Kb</td>
<td>Container bitrate for the output, or the sum of video and all audio bitrates for the output, if the container bitrate is not specified.</td>
</tr>
<tr>
<td>$\text{w}$</td>
<td>Pixels</td>
<td>Horizontal resolution.</td>
</tr>
<tr>
<td>$\text{h}$</td>
<td>Pixels</td>
<td>Vertical resolution.</td>
</tr>
<tr>
<td>$\text{f}$</td>
<td>Integer</td>
<td>FPS frame rate without decimal places. For example, “23.976” appears as “23”.</td>
</tr>
</tbody>
</table>
### Identifier Format Description

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$$</td>
<td>$</td>
<td>Escaped $.$</td>
</tr>
<tr>
<td>$sn$</td>
<td>Integer, fixed length</td>
<td>Number of the segment of the video in the output.</td>
</tr>
<tr>
<td>%0n</td>
<td>Padding modifier</td>
<td>Modifier for any data identifier. The modifier pads the resolved value with leading zeros. The format is%0n, where n is a number. For example, to ensure the resolved value in the $h$ identifier is 5 characters long, specify the identifier as $h%05$. If the vertical resolution is “720”, then the resolved, padded value is “00720”.</td>
</tr>
</tbody>
</table>

### 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.

<table>
<thead>
<tr>
<th>Object</th>
<th>Field</th>
<th>Acceptable Identifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel – Archive, HLS, Microsoft Smooth output groups</td>
<td>Destination field in an Output group</td>
<td>$dt$, $d$, $t$</td>
</tr>
<tr>
<td>Channel – Archive, Microsoft Smooth output groups</td>
<td>Name modifier field in an Output</td>
<td>All except $ra$, $rc$, $sn$</td>
</tr>
<tr>
<td>Channel – HLS output groups</td>
<td>Name modifier field in an Output</td>
<td>All except $sn$</td>
</tr>
<tr>
<td>Channel – Archive, Microsoft Smooth output groups</td>
<td>Segment modifier field in an Output</td>
<td>All except $ra$, $rc$, $sn$</td>
</tr>
<tr>
<td>Channel – HLS output groups</td>
<td>Segment modifier field in an Output</td>
<td>All except $sn$</td>
</tr>
<tr>
<td>Schedule – HLS ID3 Segment Tagging action</td>
<td>Tag field</td>
<td>All</td>
</tr>
</tbody>
</table>
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](https://aws.amazon.com/about-aws/compliance/).
- **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 (p. 345)
- Identity and Access Management in AWS Elemental MediaLive (p. 346)
- Compliance Validation for AWS Elemental MediaLive (p. 346)
- Resilience in AWS Elemental MediaLive (p. 346)
- Infrastructure Security in AWS Elemental MediaLive (p. 347)

Data Protection in AWS Elemental MediaLive

AWS Elemental MediaLive doesn't require that you supply any customer data. There are no fields in channels, inputs, or input security groups where there is an expectation that you will provide customer data.

Don't put sensitive identifying information such as customer account numbers into free-form fields such as a **Name** field. This includes when you work with MediaLive using the console, REST API, AWS CLI, or AWS SDKs. Any given piece of data that you enter into MediaLive might get picked up for inclusion in diagnostic logs.

When you provide a URL to an external server, don't include credentials information in the URL to validate your request to that server. 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.
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. 11) and Setting Up: IAM Permissions for Production (p. 18).

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,
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.

You can call these API operations from any network location, but MediaLive does support resource-based access policies, which can include restrictions based on the source IP address. You can also use MediaLive policies to control access from specific Amazon Virtual Private Cloud (Amazon VPC) endpoints or specific VPCs. Effectively, this isolates network access to a given MediaLive resource from only the specific VPC within the AWS network.
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. 8).

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. Unless otherwise noted, there is a separate quota for each Region in each Amazon Web Services account.

<table>
<thead>
<tr>
<th>Resource or Operation</th>
<th>Default Quota</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channels</td>
<td>5</td>
<td>The maximum number of channels that you can create.</td>
</tr>
<tr>
<td>Channels with HEVC</td>
<td>5</td>
<td>The maximum number of channels that include one or more HEVC outputs that you can create.</td>
</tr>
<tr>
<td>Channels with UHD</td>
<td>1</td>
<td>The maximum number of channels that include a UHD output that you can create in this account in the current Region. For information about the maximum number of UHD outputs in these channels, see Feature Rules and Limits (p. 8).</td>
</tr>
<tr>
<td>Inputs of type push (not including VPC push inputs)</td>
<td>5</td>
<td>The maximum number of push inputs (not including VPC push inputs) that you can create in the current Region.</td>
</tr>
<tr>
<td>Inputs of type pull</td>
<td>100</td>
<td>The maximum number of pull inputs that you can create.</td>
</tr>
<tr>
<td>Inputs of type VPC push</td>
<td>50</td>
<td>The maximum number of VPC push inputs that you can create.</td>
</tr>
<tr>
<td>Input security groups</td>
<td>5</td>
<td>The maximum number of input security groups that you can create.</td>
</tr>
<tr>
<td>Multiplexes</td>
<td>2</td>
<td>The maximum number of channels that you can create in the current Region.</td>
</tr>
<tr>
<td>Reservations</td>
<td>50</td>
<td>The maximum number of reservations that you can create.</td>
</tr>
</tbody>
</table>
The following table describes the documentation for this release of AWS Elemental MediaLive

- **API version:** latest

<table>
<thead>
<tr>
<th>update-history-change</th>
<th>update-history-description</th>
<th>update-history-date</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCTE-35 schedule actions in follow mode (p. 159)</td>
<td>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.</td>
<td>April 24, 2020</td>
</tr>
<tr>
<td>Revision to documentation for creating output groups (p. 80)</td>
<td>The procedures for creating an output group and an output have been revised and reorganized.</td>
<td>April 23, 2020</td>
</tr>
<tr>
<td>Redundant HLS manifests (p. 285)</td>
<td>There is a new section in the guide to describe the recently added support for redundant HLS manifests.</td>
<td>April 23, 2020</td>
</tr>
<tr>
<td>Custom manifests in HLS output groups (p. 280)</td>
<td>There is a new section in the guide to describe the existing feature of customizing the paths for HLS manifests.</td>
<td>April 23, 2020</td>
</tr>
<tr>
<td>Section for changing channel class (p. 122)</td>
<td>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.</td>
<td>April 3, 2020</td>
</tr>
<tr>
<td>Resiliency section (p. 298)</td>
<td>The MediaLive Features section now includes a subsection that lists the various features that let you improve resiliency in the MediaLive channel.</td>
<td>April 3, 2020</td>
</tr>
<tr>
<td>New feature—Automatic input failure (p. 218)</td>
<td>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.</td>
<td>April 3, 2020</td>
</tr>
<tr>
<td>Channel class (p. 243)</td>
<td>The MediaLive Features section now includes a subsection about</td>
<td>April 3, 2020</td>
</tr>
<tr>
<td>Feature/Changes</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>Timecode configuration (p. 321)</td>
<td>The user guide now includes information on configuring timecode, including the supported standards for timecodes in the input and for timecodes in the output.</td>
<td>March 26, 2020</td>
</tr>
<tr>
<td>Audio in an HLS input (p. 339)</td>
<td>We have added one line to clarify that MediaLive doesn't support ingest of audio in rendition groups.</td>
<td>March 26, 2020</td>
</tr>
<tr>
<td>Audio rendition groups for HLS outputs (p. 210)</td>
<td>There is a new section in the guide to describe the existing feature of audio rendition groups in HLS outputs.</td>
<td>March 3, 2020</td>
</tr>
<tr>
<td>Input switching limitations (p. 264)</td>
<td>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.</td>
<td>February 2, 2020</td>
</tr>
<tr>
<td>Number of frame capture outputs (p. 341)</td>
<td>A channel can now contain up to three frame capture output groups. Previously, it could contain only one.</td>
<td>January 31, 2020</td>
</tr>
<tr>
<td>New variable data identifier: $sn$</td>
<td>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.</td>
<td>December 31, 2019</td>
</tr>
<tr>
<td>ID3 segment tagging (p. 258)</td>
<td>AWS Elemental MediaLive now supports the ability to insert ID3 tags in every segment in HLS and MediaPackage outputs.</td>
<td>December 31, 2019</td>
</tr>
<tr>
<td>Summary of rules and constraints (p. 8)</td>
<td>The user guide now includes a section that describes some of the rules and constraints that apply to AWS Elemental MediaLive.</td>
<td>December 12, 2019</td>
</tr>
<tr>
<td>IAM access requirements for multiplexes (p. 23)</td>
<td>The list of operations required for the multiplex feature has been revised to include ec2.DescribeAvailabilityZones.</td>
<td>December 12, 2019</td>
</tr>
<tr>
<td>Topic</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Nielsen watermarks (p. 297)</td>
<td>AWS Elemental MediaLive now supports the ability to convert Nielsen watermarks to ID3 metadata.</td>
<td>December 9, 2019</td>
</tr>
<tr>
<td>Revision to documentation for schedules (p. 341)</td>
<td>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.</td>
<td>December 8, 2019</td>
</tr>
<tr>
<td>SCTE-35 schedule actions in immediate mode (p. 159)</td>
<td>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.</td>
<td>November 25, 2019</td>
</tr>
<tr>
<td>SCTE-35 schedule actions in immediate mode (p. 341)</td>
<td>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.</td>
<td>November 25, 2019</td>
</tr>
<tr>
<td>Permissions for MediaLive multiplex (p. 23)</td>
<td>The information about requirements for user access has been updated to include access to the multiplex feature.</td>
<td>November 25, 2019</td>
</tr>
<tr>
<td>Multiplex and MPTS (p. 290)</td>
<td>You can now create a multiplex to produce an MPTS.</td>
<td>November 25, 2019</td>
</tr>
<tr>
<td>Color space (p. 246)</td>
<td>AWS Elemental MediaLive now includes options for handling video color space metadata.</td>
<td>September 12, 2019</td>
</tr>
<tr>
<td>Ad avails: Manifest decoration in HLS (p. 303)</td>
<td>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.)</td>
<td>September 12, 2019</td>
</tr>
<tr>
<td>Support for an immediate input switch (p. 160)</td>
<td>The channel schedule now supports the ability to enter a request to switch an input as soon as possible (immediate mode).</td>
<td>July 25, 2019</td>
</tr>
<tr>
<td>Input clipping (p. 262)</td>
<td>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.</td>
<td>July 25, 2019</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Dynamic inputs (p. 254)</td>
<td>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.</td>
<td>July 25, 2019</td>
</tr>
<tr>
<td>Security information (p. 345)</td>
<td>The new Security chapter contains information about how to configure AWS Elemental MediaLive to meet your security and compliance objectives.</td>
<td>May 28, 2019</td>
</tr>
<tr>
<td>Reorganization of the user guide (p. 1)</td>
<td>The chapters in the user guide have been reorganized and renamed. No sections have been removed. The &quot;components&quot; 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 &quot;MediaLive Features&quot;.</td>
<td>May 28, 2019</td>
</tr>
<tr>
<td>Quotas for VPC inputs (p. 348)</td>
<td>The table for MediaLive quotas now includes the quotas for VPC inputs.</td>
<td>May 3, 2019</td>
</tr>
<tr>
<td>Changing the channel class (p. 121)</td>
<td>You can now change the channel class for an existing channel.</td>
<td>May 3, 2019</td>
</tr>
<tr>
<td>Changing the channel class (p. 122)</td>
<td>You can now change the channel class for an existing channel.</td>
<td>May 3, 2019</td>
</tr>
<tr>
<td>Standard channels and single-pipeline channels (p. 71)</td>
<td>You can now set up a channel with a single-pipeline. The Create channel page includes a Channel class field.</td>
<td>April 5, 2019</td>
</tr>
<tr>
<td>Revision to documentation for setting up access (p. 18)</td>
<td>The procedures for setting up the user and the service with AWS IAM access have been revised and split into two chapters, Setting Up&gt; and Setting Up for Production.</td>
<td>March 22, 2019</td>
</tr>
<tr>
<td>MediaPackage output group (p. 103)</td>
<td>There is a new output group type: MediaPackage, which lets you send output to AWS Elemental MediaPackage. This new output group type creates an HLS output that is streamlined for delivery to MediaPackage.</td>
<td>March 13, 2019</td>
</tr>
<tr>
<td>Support for pausing and unpauseing a channel pipeline (p. 161)</td>
<td>The channel schedule now supports the ability to enter actions to pause and unpause a pipeline in a channel.</td>
<td>March 8, 2019</td>
</tr>
<tr>
<td>Feature Description</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>----------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>Support for encrypted HLS inputs (p. 61)</td>
<td>MediaLive now supports ingest of encrypted HLS inputs.</td>
<td>March 5, 2019</td>
</tr>
<tr>
<td>Push inputs from your VPC (p. 126)</td>
<td>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.</td>
<td>February 20, 2019</td>
</tr>
<tr>
<td>Quotas for reservations (p. 348)</td>
<td>The table for MediaLive quotas now includes the quotas for reservations.</td>
<td>February 11, 2019</td>
</tr>
<tr>
<td>Tagging (p. 319)</td>
<td>There is now support for attaching tags to channels, inputs, and input security groups.</td>
<td>February 8, 2019</td>
</tr>
<tr>
<td>Frame capture output group (p. 86)</td>
<td>There is a new output group type: frame capture output group, to send a series of frame capture files to Amazon Simple Storage Service.</td>
<td>January 25, 2019</td>
</tr>
<tr>
<td>Integration with AWS CloudTrail (p. 207)</td>
<td>There is now support for logging MediaLive API calls with CloudTrail.</td>
<td>January 18, 2019</td>
</tr>
<tr>
<td>Integration with AWS Elemental MediaConnect (p. 126)</td>
<td>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.</td>
<td>December 7, 2018</td>
</tr>
<tr>
<td>Revision to documentation for schedules (p. 159)</td>
<td>The channel schedule now supports the ability to enter actions to switch the channel from ingest one input to another input. This chapter has been revised.</td>
<td>November 8, 2018</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>-------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Requirement for redundant channels in AWS Elemental MediaPackage</td>
<td>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. 50) and the section called “Setting up the downstream system” (p. 65) have been revised to reflect this change.</td>
<td>August 31, 2018</td>
</tr>
<tr>
<td>Schedule feature</td>
<td>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 MediaLive Schedule (p. 159) and the section called “Image Overlays” (p. 260) have been added.</td>
<td>August 24, 2018</td>
</tr>
<tr>
<td>Reservations feature</td>
<td>There is now support for purchasing a reservation for</td>
<td>June 19, 2018</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>Channel logs feature</td>
<td>There is now support for sending log information to Amazon CloudWatch Logs. The sections the section called “Logging” (p. 80) and the section called “Monitoring a Channel Using CloudWatch Logs” (p. 205) have been added. The section Setting Up: IAM Permissions for Production (p. 18) has been revised to include setup for logs.</td>
<td>June 13, 2018</td>
</tr>
<tr>
<td>Console alerts feature</td>
<td>There is now support for viewing channel alerts on the console. See Monitoring a Channel or Multiplex (p. 199). The information on setting up for alerts has been moved to this chapter.</td>
<td>June 6, 2018</td>
</tr>
<tr>
<td>Support for RTMP outputs</td>
<td>the section called “Setting up the downstream system” (p. 65) now includes RTMP. The section the section called “RTMP output group” (p. 111) has been added. the section called “Captions Categories” (p. 327) now includes RTMP CaptionInfo. the section called “Formats Supported in an RTMP Output” (p. 333) has been added.</td>
<td>April 18, 2018</td>
</tr>
<tr>
<td>Input edit feature, Input security group edit feature</td>
<td>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. 142) the section called “Editing an Input Security Group” (p. 146)</td>
<td>March 23, 2018</td>
</tr>
<tr>
<td>Input delete feature, Input security group delete feature</td>
<td>Deleting an input and deleting an input security group are now supported. See the section called “Editing an input” (p. 142) the section called “Editing an Input Security Group” (p. 146)</td>
<td>March 23, 2018</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Creating a channel</td>
<td>the section called “Creating a channel from scratch” (p. 69) has been extensively revised, particularly with more information and examples on setting up destinations for output groups.</td>
<td>March 23, 2018</td>
</tr>
<tr>
<td>Input specification feature</td>
<td>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. 73).</td>
<td>February 15, 2018</td>
</tr>
<tr>
<td>Channel edit feature</td>
<td>Editing of the fields in an existing (saved) channel is now supported. See the section called “Editing a Channel” (p. 121).</td>
<td>February 15, 2018</td>
</tr>
<tr>
<td>Custom template feature</td>
<td>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. 118).</td>
<td>February 15, 2018</td>
</tr>
<tr>
<td>New service and guide</td>
<td>This is the initial release of AWS Elemental MediaLive User Guide.</td>
<td>November 27, 2017</td>
</tr>
</tbody>
</table>

**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.