User Guide

MediaConvert

MediaConvert: User Guide

Table of Contents

What is AWS Elemental MediaConvert?

AWS Elemental MediaConvert is a file-based video processing service that provides scalable video processing for content owners and distributors with media libraries of any size. MediaConvert offers advanced features that enable premium content experiences, including:

- Professional broadcast codecs that support increased bit depth and HDR content creation
- Still graphic overlays
- Advanced audio
- Digital rights management (DRM)
- Closed captioning support

MediaConvert offers support for various input formats and adaptive bitrate (ABR) packaging output formats for delivering high-quality content from a range of sources onto primary and multiscreen devices.

For simple use cases, you can set up a MediaConvert transcoding job in a few steps. For instructions, see Getting started with MediaConvert.

MediaConvert has the following components:

Jobs

A *job* does the work of transcoding. Each job converts an input file into an output file or files. Inputs and outputs can contain one or more of video, audio, and captions, either together or in separate files. Before you begin creating jobs, make sure that you know what your input files are and what they contain. Also make sure that you know what files you would like to create as outputs and what format you would like them in.

When you create a job, you specify the name of the file that you want to transcode, the names that you want MediaConvert to give to the finished output files, and several other settings. For more information, see Working with jobs.

Presets

A *preset* is a saved group of encoding settings for a single output. You can create many common outputs by simply selecting a system preset. You can also create your own custom presets, either by duplicating and modifying an existing preset or by creating one from scratch.

When you create a job, you can specify a preset that you want to use, or you can individually specify your encoding settings. For more information, see Working with output presets.

Job templates

A *job template* specifies all the settings for a complete job, except for your IAM role and those settings that are likely to change for each job, such as the input file location and name, and user metadata that you might tag the job with. You create a job template by specifying all input settings other than input location and file name, and then specifying all the outputs that the job will generate. You can specify the settings for each output by choosing a preset for the output or by specifying each output setting individually. For more information, see Working with job templates.

Queues

Use *queues* to manage the resources that are available to your account for parallel processing of jobs. For more information, see Working with queues in AWS Elemental MediaConvert.

Setting up

AWS Elemental MediaConvert transcodes your input files from Amazon S3, HTTP, or HTTPS locations to generate output files on Amazon S3. To use MediaConvert, you need an AWS account, at least one input file, an Amazon S3 bucket for your output files, and an IAM role with the correct permissions.

For information on how to upload files to Amazon S3, see Uploading objects in the *Amazon S3 User Guide*.

For information about creating an Amazon S3 bucket for your output destination, see Creating a bucket in the *Amazon S3 User Guide*.

The following topics describe how to sign up for an AWS account and then how to configure your IAM role.

Topics

Sign up for an AWS account

When you sign up for AWS, your AWS account is automatically signed up for all services in AWS, including AWS Elemental MediaConvert. You are charged only for the services that you use.

With MediaConvert, you pay only for what you use. For more information about MediaConvert features and pricing, see MediaConvert.

When you sign up for AWS and set up MediaConvert, you can optionally change the display language in the AWS Management Console. For more information, see Changing the language of the AWS Management Console in the *AWS Management Console Getting Started Guide*.

Topics

Sign up for an AWS account

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

To sign up for an AWS account

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

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

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

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

Create a user with administrative access

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

Secure your AWS account root user

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

   For help signing in by using root user, see Signing in as the root user in the *AWS Sign-In User Guide*.
2. Turn on multi-factor authentication (MFA) for your root user.

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

Create a user with administrative access

1. Enable IAM Identity Center.

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

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

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

Sign in as the user with administrative access

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

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

Assign access to additional users

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

   For instructions, see Create a permission set in the *AWS IAM Identity Center User Guide*.

2. Assign users to a group, and then assign single sign-on access to the group.

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

Setting up IAM permissions

To run transcoding jobs with AWS Elemental MediaConvert, you need an IAM service role to allow MediaConvert access to your resources. Resources include things like your input files and the locations where your output files are stored.

Regardless of how you initially create your IAM service role, you can refine this role at any time using IAM. For more information, see Adding and removing IAM identity permissions in the *IAM User Guide*.

You can create your IAM service role in one of the following ways:

- In the MediaConvert console, with some restrictions on the permissions that you grant. For instructions, see Creating the IAM role in MediaConvert.

  From the MediaConvert console, by configuring your role to allow MediaConvert access to only some of your Amazon S3 buckets. You can also choose whether to grant invoke access to your API Gateway endpoints.
- In the IAM console. For instructions, see Creating a role in IAM.

  You can exercise fine control over exactly what access you grant to MediaConvert when you set up your IAM role in the IAM console. You can also use IAM through the AWS Command Line Interface (AWS CLI), or an API or SDK.

> **Note**
>
> If you enable Amazon S3 default encryption on your Amazon S3 buckets, and you and specify your own key managed by AWS Key Management Service, you must grant additional permissions. For more information, see Granting permissions for MediaConvert to access encrypted Amazon S3buckets.

Set up a default role

If you use the name `MediaConvert_Default_Role`, then the MediaConvert console uses it by default when you create jobs in the future. This happens regardless of how you create the IAM service role for MediaConvert to use.

Creating the IAM role in MediaConvert with configured permissions

When you create the AWS Identity and Access Management (IAM) role in MediaConvert with configured permissions, you can restrict MediaConvert access to only specific Amazon S3 buckets. You can also specify whether to grant invoke access to your Amazon API Gateway endpoints.

To set up the IAM role in MediaConvert with configured permissions

1. Open the [Jobs](#) page in the MediaConvert console.
2. Choose **Create job**.
3. Under **Job settings**, choose **AWS integration**.
4. In the **Service access** section, for **Service role control**, choose **Create a new service role, configure permissions**.
5. For **New role name**, we suggest that you keep the default value **`MediaConvert_Default_Role`**. When you do, MediaConvert uses this role by default for your future jobs.
6. For **Input S3 locations** and **Output S3 locations**, choose **Add location**. Select the Amazon S3 buckets that you will use for input or output locations.
7. (Optional) For **API Gateway endpoint invocation**, if you use features that require it, choose allow.

   MediaConvert requires this access for the following features:

   - Digital rights management with SPEKE
   - Nielsen non-linear watermarking

   To allow MediaConvert invoke access to a specific endpoint only, modify these permissions in the role policy after you create it by using the AWS Identity and Access Management (IAM) service. For more information, see [Editing IAM policies](#) in the *AWS Identity and Access Management User Guide*.

Creating a role with the IAM console

Working directly with AWS Identity and Access Management (IAM), you can do actions that aren't available in the MediaConvert console. You can either do this when you create your role in IAM, or you can create your role in MediaConvert and then use IAM to refine it later.

The following procedure explains how to create a role with the IAM console. For information about accessing IAM programmatically, see the appropriate document in the [IAM documentation set](#).

To create the service role for AWS Elemental MediaConvert (IAM console)

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane of the IAM console, choose **Roles**, and then choose **Create role**.
3. For **Trusted entity type**, choose **AWS service**.
4. For **Service or use case**, choose **AWS Elemental MediaConvert**, and then choose the **MediaConvert** use case.
5. Choose **Next**.
6. Select the box next to the MediaConvert policy that you created in the previous procedure.
7. (Optional) Set a permissions boundary. This is an advanced feature that is available for service roles, but not service-linked roles.

    a. Open the **Set permissions boundary** section, and then choose **Use a permissions boundary to control the maximum role permissions**.

       IAM includes a list of the AWS managed and customer-managed policies in your account.

    b. Select the policy to use for the permissions boundary.
8. Choose **Next**.
9. Enter a role name or a role name suffix to help you identify the purpose of the role.

> **Important**
>
> When you name a role, note the following:
>
> - Role names must be unique within your AWS account, and can't be made unique by case.
>
>   For example, don't create roles named both `PRODROLE` and `prodrole`. When a role name is used in a policy or as part of an ARN, the role name is case sensitive, however when a role name appears to customers in the console, such as during the sign-in process, the role name is case insensitive.
> - You can't edit the name of the role after it's created because other entities might reference the role.

10. (Optional) For **Description**, enter a description for the role.

11. (Optional) To edit the use cases and permissions for the role, in the **Step 1: Select trusted entities** or **Step 2: Add permissions** sections, choose **Edit**.
12. (Optional) To help identify, organize, or search for the role, add tags as key-value pairs. For more information about using tags in IAM, see Tagging IAM resources in the *IAM User Guide*.
13. Review the role, and then choose **Create role**.

> **Note**
>
> For **New role name**, we suggest that you enter **`MediaConvert_Default_Role`**. When you do, MediaConvert uses this role by default for your future jobs.

Granting permissions for MediaConvert to access encrypted Amazon S3buckets

When you enable Amazon S3 default encryption, Amazon S3 automatically encrypts your objects as you upload them. You can optionally choose to use AWS Key Management Service (AWS KMS) to manage the key. This is called SSE-KMS encryption.

If you enable SSE-KMS default encryption on the buckets that hold your AWS Elemental MediaConvert input or output files, you must add inline policies to your IAM service role. If you don't add inline policies, MediaConvert can't read your input files or write your output files.

Grant these permissions in the following use cases:

- If your input bucket has SSE-KMS default encryption, grant `kms:Decrypt`.
- If your output bucket has SSE-KMS default encryption, grant `kms:GenerateDataKey`.

The following example inline policy grants both permissions.

Example inline policy with kms:Decrypt and kms:GenerateDataKey

This policy grants permissions for both `kms:Decrypt` and `kms:GenerateDataKey`.

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

```
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike":

{            "kms:ViaService": "s3.*.amazonaws.com"         }
      }
    }
  ]
}
```

Getting started with MediaConvert

To get started using the MediaConvert console, this tutorial shows how to create a job to transcode media files. To access MediaConvert programmatically, see the following topics in the API Reference:

- If you are using one of the AWS SDKs, see Getting started with the SDKs.
- If you are using the MediaConvert API directly, see Getting started with the API.

MediaConvert takes in an input file and turns it into one or more output files, based on the instructions and transcoding settings that you provide.

> **Note**
>
> If you aren't familiar with MediaConvert basics like jobs, queues, presets, and job templates, read What is AWS Elemental MediaConvert?.

Topics

Prerequisites

Follow the steps in the Setting up chapter so that your input files are accessible and MediaConvert has permissions to run your job.

Start by noting the location of your input files. This will be a URI like *s3://DOC-EXAMPLE-BUCKET/input.mp4* or a URL like *https://example.amazon.com/input.mp4*. Then, note the location of your Amazon S3 destination for your output files. You will use this input and output information when you create your job.

For more information about which input and output formats MediaConvert supports, see Supported inputs and outputs.

Create a job

A job does the work of transcoding. When you create a job, you specify the input files and settings, the output files and settings, and any other related job settings.

MediaConvert gets the input from the Amazon S3, HTTP, or HTTPS location that you specify. Then MediaConvert transcodes and writes to the output location that you specify in the job's output group settings.

To create a job

1. Go to the Jobs page in the MediaConvert console.
2. Choose **Create job**.
3. On the **Create job** page, specify your job settings. For detailed information, see Configuring jobs in MediaConvert.

   Make sure that you select the same Region for your job and your file storage.
4. Choose **Create**.

   For information about tracking the status of your job, see Using EventBridge with AWS Elemental MediaConvert.

   For information about the file names and paths for your job outputs, see Output file names and paths.
5. Optionally, if you don't want to keep the transcoded files that you generate during this tutorial, delete them from Amazon S3 to avoid incurring storage charges. For more information, see Deleting objects *Amazon S3 User Guide*.

Supported inputs and outputs

This chapter contains reference tables and other detailed information for inputs and outputs that AWS Elemental MediaConvert supports.

Topics

Supported input formats

AWS Elemental MediaConvert supports inputs with the following containers and codecs.

Some containers and codecs have additional restrictions. For more information about the video container, video codec, or audio codec, choose its link in the following reference table, or see Supported containers and codecs reference tables.

Container	Supported video codecs	Supported audio codecs
3G2, 3GP	AVC (H.264) H.263 MPEG-4 Part 2	AAC AMR-NB AMR-WB
ASF, WMV (Advanced Systems Format)	VC-1	WMA WMA2 WMA Pro
AVI (Audio Video Interleave)	Canopus HQ DivX/Xvid DV/DVCPRO	Dolby Digital (AC3) Dolby Digital Plus (EAC3) Dolby E

Container	Supported video codecs	Supported audio codecs
	MJPEG Uncompressed	MP3 MPEG Audio PCM
F4V, FLV (MPEG-4 Flash)	AVC (H.264) H.263	AAC
GIF	GIF	*Not applicable*
HLS (Apple HTTP Live Streaming) With MPEG-TS segments	AVC (H.264) HEVC (H.265) MPEG-2	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3)
IMF (Interoperable Master Format)	Apple ProRes JPEG 2000 (J2K)	PCM
Matroska	AVC (H.264) MPEG-2 MPEG-4 Part 2 VC-1	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3) Opus WMA WMA2 PCM FLAC

Container	Supported video codecs	Supported audio codecs
MOV (Apple QuickTime)	Apple ProRes AVC (H.264) AVC-Intra DivX/Xvid DV/DVCPRO H.261 H.262 H.263 HEVC (H.265) JPEG 2000 (J2K) MJPEG MPEG-2 MPEG-4 Part 2 QuickTime Animation (RLE) Uncompressed	AAC MP3 PCM

Container	Supported video codecs	Supported audio codecs
MP4 (MPEG-4)	AV1 AVC (H.264) AVC-Intra DivX/Xvid H.261 H.262 H.263 HEVC (H.265) JPEG 2000 (J2K) MPEG-2 MPEG-4 Part 2 VC-1 VP9 Uncompressed	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3) PCM WMA WMA2
MPEG-1 (MPEG-1 System Stream)	MPEG-1 MPEG-2	AAC AIFF Dolby Digital (AC3) Dolby Digital Plus (EAC3) MPEG Audio PCM

Container	Supported video codecs	Supported audio codecs
MPEG-PS (MPEG Program Stream)	MPEG-2	MPEG Audio
MPEG-TS (MPEG-2 Transport Stream)	AVC (H.264) HEVC (H.265) MPEG-2 VC-1	AAC AIFF Dolby Digital (AC3) Dolby Digital Plus (EAC3) Dolby E MPEG Audio PCM WMA WMA2

Container	Supported video codecs	Supported audio codecs
MXF (Material Exchange Format)	Apple ProRes AVC (H.264) AVC-Intra DV/DVCPRO DV25 DV50 DVCPro HD JPEG 2000 (J2K) MPEG-2 Panasonic P2 Sony XDCAM Sony XDCAM MPEG-4 Proxy VC-3 Uncompressed	AAC AIFF Dolby E MPEG Audio PCM
WebM	VP8 VP9	Opus Vorbis
No container	AVC (H.264) DV/DVCPRO HEVC (H.265) MPEG-1 MPEG-2	FLAC PCM

Supported audio-only workflow input formats

When you create audio-only outputs, MediaConvert supports audio-only inputs with the following containers and codecs.

Some containers and codecs have additional restrictions. For more information about the audio container or audio codec choose its link in the following reference table. Or see Supported containers and codecs reference tables.

Container	Supported audio Codecs
ASF, WMA, WMV (Advanced Systems Format)	WMA WMA2 WMA Pro
FLAC (Free Lossless Audio Codec)	FLAC
Matroska	FLAC Opus
MOV (Apple QuickTime)	PCM
MP3 (MPEG-1 Layer 3)	MP3
MP4 (MPEG-4)	AAC
MPEG-TS (MPEG-2 Transport Stream)	MPEG-2 PCM
OGG, OGA	FLAC Opus Vorbis
WAV	PCM
No container	AAC

Container	Supported audio Codecs
	Dolby Digital (AC3) Dolby Digital Plus (EAC3) FLAC PCM

Using HLS inputs with MediaConvert

When your input to MediaConvert is an HLS package, specify either a parent or child manifest for **Input file URL** (`FileInput`). When the manifest is a parent that lists multiple child manifests, MediaConvert uses the child manifest with the highest bandwidth as the input source.

Features compatible with HLS inputs

With HLS inputs, you can use the following input features:

- Input clipping
- Input stitching
- Image insertion
- Embedded input captions selectors

Feature restrictions with HLS inputs

When your input is an HLS package, your job is restricted in these ways:

- Your input package must conform to the requirements listed in HLS input package requirements.
- Your input segments can't be encrypted with DRM. For example, your inputs can't be encrypted with Apple FairPlay DRM.
- You can use only embedded input captions.

HLS input package requirements

Your input HLS package must conform to the following requirements:

- The video container for your media segments must be MPEG-2 TS.
- The compatibility version of the manifest file (specified by `EXT-X-VERSION`) must be 4 or smaller.
- The manifest file must stay the same after you submit your job. That is, the manifest must have the tag `EXT-X-ENDLIST` or it must have the `EXT-X-PLAYLIST-TYPE` value set to VOD.
- If the manifest uses `EXT-X-BYTERANGE`, the start of the first subrange must be 0 and the following subrange segments must continue the former one.
- If the input has discontinuities, they must start at the beginning of a segment. That is, the input can't have discontinuities in the subrange of a segment.
- If the manifest includes `EXT-X-KEY`, `METHOD` must be set to `NONE`. For example: `EXT-X-KEY:METHOD=NONE`. MediaConvert does not support HLS encrypted inputs.
- MediaConvert ignores the following tags:
  - `EXT-X-PROGRAM-DATE-TIME`
  - `EXT-X-DATERANGE`
  - `EXT-X-I-FRAMES-ONLY`
  - `EXT-X-I-FRAME-STREAM-INF`
  - `EXT-X-SESSION-DATA`
  - `EXT-X-SESSION-KEY`
  - `EXT-X-INDEPENDENT-SEGMENTS`
  - `EXT-X-START`
- When your job uses accelerated transcoding, your input HLS package must conform to this additional requirement: The duration in `EXTINF` must be specified using a decimal floating-point, with enough accuracy to avoid perceptible errors when segment durations are accumulated.

Using alternate audio renditions

With HLS rendition groups, you can use the audio selector settings to indicate which alternate audio rendition you want MediaConvert to use. To be eligible for selection, your alternate audio renditions must conform to the following requirements:

- The renditions must be included in `EXT-X-MEDIA` tags in the input parent manifest.

- The `EXT-X-MEDIA` tags must contain a unique combination of GROUP-ID, NAME, and LANGUAGE values..
- Audio must be in one of the following supported audio codecs: AAC, Dolby Digital (AC3), Dolby Digital Plus (EAC3), or MP3.
- The child manifest for your alternate audio rendition must be included in the parent manifest that you used for your Input file URL (FileInput)

When you specify audio selector settings to identify an alternate audio rendition, the audio selector looks for a matching `EXT-X-MEDIA` tag in the parent manifest.

You can use one or more selector settings at a time. For example, given the following `EXT-X-MEDIA` tags, you can identify the audio rendition by the name (RenditionName) or language (RenditionLangageCode) because these are both unique values across the tags.

```
#EXT-X-MEDIA:TYPE=AUDIO,GROUP-
ID="audio",CHANNELS="2",NAME="English",LANGUAGE="eng",DEFAULT=YES,AUTOSELECT=YES
```

```
#EXT-X-MEDIA:TYPE=AUDIO,GROUP-
ID="audio",CHANNELS="2",NAME="Japanese",LANGUAGE="jpn",DEFAULT=NO,AUTOSELECT=NO,
```

However, because the group ID (RenditionGroupID) is the same for both tags, you can't use that alone to identify an audio rendition. You must use the group ID in combination with another value from the `EXT-X-MEDIA` tag to identify the audio rendition that you want MediaConvert to use.

If you don't specify audio selector settings, the audio selector looks for audio that's muxed into the video segments. If the video segments don't contain audio, the audio selector uses the first alternate audio rendition from the input parent manifest.

HTTP input requirements

When your input file source is HTTP(S), you specify the URL rather than an Amazon S3 path. Requirements for using HTTP(S) for input are as follows:

- All input files must be publicly readable.
- The HTTP(S) server must not require authentication.
- The HTTP(S) server must accept both HEAD and range GET requests.
- The URL that you specify can't include parameters.

If your HTTP(S) input uses redirection, it must follow these restrictions:

- You can redirect only once from the URL that you provide as your input. You can't redirect to a URL that, in turn, contains a redirect.
- The HTTP(S) status response code from the initial server must be either 301 or 302.
- The HTTP(S) response from the initial server must use its `Location` headers to provide the URL that it's redirecting MediaConvert to.

Supported output formats

MediaConvert supports the following combinations of output containers and codecs.

For outputs that contain *only* audio inside the output container, MediaConvert supports a different set of containers and codecs. For more information, see Audio-only.

Some containers and codecs have additional restrictions. For more information about the video container, video codec, or audio codec, choose its link in the following reference table. Or see Supported containers and codecs reference tables.

Container	Supported video codecs	Supported audio codecs
CMAF DASH (Common Media Application Format)	AV1 AVC (H.264) HEVC (H.265)	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3)
CMAF HLS (Common Media Application Format)	AV1 AVC (H.264) HEVC (H.265)	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3)
DASH (Dynamic Adaptive Streaming over HTTP)	AV1 AVC (H.264) HEVC (H.265) VP8	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3)

Container	Supported video codecs	Supported audio codecs
	VP9	
F4V (MPEG-4 Flash)	AVC (H.264) MPEG-2	AAC
HLS (Apple HTTP Live Streaming)	AVC (H.264) HEVC (H.265)	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3)
MOV (Apple QuickTime)	AVC (H.264) MPEG-2	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3) WAV
	Apple ProRes	AIFF
MP4 (MPEG-4)	AV1 AVC (H.264) HEVC (H.265)	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3)
MPEG-TS (MPEG-2 Transport Stream)	AVC (H.264) HEVC (H.265) MPEG-2	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3) MPEG-2 PCM/WAV

Container	Supported video codecs	Supported audio codecs
MSS (Microsoft Smooth Streaming	AVC (H.264)	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3)
MXF (Material Exchange Format)	AVC (H.264) AVC-Intra MPEG-2 VC-3 XAVC	PCM/WAV
WebM	VP8 VP9	Opus Vorbis
Y4M	Uncompressed	*Not supported*
No container	AVC-Intra AVC (H.264) HEVC (H.265) MPEG-2 VC-3 XAVC	AAC AIFF Dolby Digital (AC3) Dolby Digital Plus (EAC3) PCM/WAV

Audio-only

MediaConvert supports the following combinations of output container and codec for creating audio-only outputs.

Some containers and codecs have additional restrictions. For more information about the audio container or audio codec choose its link in the following reference table. Or see Supported containers and codecs reference tables.

Container	Audio Codecs
DASH (Dynamic Adaptive Streaming over HTTP)	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3) Dolby Digital Plus JOC (Atmos)
HLS (Apple HTTP Live Streaming)	AAC Dolby Digital (AC3)
MP4 (MPEG-4)	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3)
MPEG-TS (MPEG-2 Transport Stream)	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3) MPEG-2
No container	AAC AIFF Dolby Digital (AC3) Dolby Digital Plus (EAC3) FLAC MPEG-2

Container	Audio Codecs
	MP3 WAV

Supported output resolution maximums by codec

The following table shows the maximum output resolution AWS Elemental MediaConvert supports for each output codec.

Codec	Maximum resolution (pixels)
AV1	4096x2160
AVC-Intra	1920x1080 or 1080x1920
AVC (H.264)	4096x2160 or 2160x4096
HEVC (H.265)	8192x4320 or 4320x8192
MPEG-2	1920x1152
Apple ProRes	4096x4096
VC-3	1920x1080 or 1080x1920
VP8, VP9	4096x2160 or 2160x4096
XAVC	4096x2160

8k output resolution job restrictions

When your MediaConvert job has outputs with 8k (8192x4320) resolutions, your job is restricted in these ways:

- You can't create Dolby Vision outputs.
- You must send your job to an on-demand queue. Reserved queues can't run 8k jobs.

Creating MXF outputs with AWS Elemental MediaConvert

MXF is an output container format that carries video content for editing, archiving, and exchange. The MXF format is governed by a set of specifications, some of which define *MXF profiles*, also called shims. These MXF profiles lay out constraints on encoding settings including video codec, resolution, and bitrate.

To make sure that your outputs comply with these specifications, you can use the MediaConvert automatic profile selection. When you do that, MediaConvert automatically encodes the correct profile, based on the values you choose for your codec, resolution, and bitrate. For more information, see [Default automatic selection of MXF profiles](#).

You can also explicitly choose your MXF profile. When you do so in the MediaConvert console, MediaConvert automatically populates the dropdown list for **Video codec** with only valid codecs. When you aren't using automatic profile selection, refer to the relevant specifications for constraints on your resolution and bitrate.

> **Note**
>
> When you manually specify your MXF profile, you must set up your output in a way that is compatible with that specification. You can submit jobs with incompatible MXF profiles and encoding settings, but those jobs will fail.

Topics

MXF job limitations and requirements

MediaConvert restricts MXF jobs in these ways:

- You can put MXF outputs in a **File group** output group only.

- You must choose a video codec that is supported with your MXF profile. The following table details which codecs are supported with each profile. For more information, see Codecs supported with each MXF profile.
- You must set up your output audio tracks according to the requirements of the MXF profile. This applies whether you specify the profile or have MediaConvert automatically select it for you. For more information, see Output audio requirements for each MXF profile.

Setting up an MXF output

To set up a job with an MXF output, you create the output in the **File group** output group. Then specify MXF as the output's container.

To set up your transcoding job with an MXF output (console)

1. Set up your job inputs and outputs as described in Configuring jobs in MediaConvert. Put your MXF output in the **File group** output group.
2. On the **Create job** page, in the **Job** pane on the left, choose your output.
3. In the **Output settings** section, for **Container**, choose **MXF container**.
4. Optionally, specify values for the settings under **MXF container settings**:
   - **MXF profile**: To manually specify your MXF profile, choose it from the list. If you keep the default value, **Auto**, MediaConvert chooses your MXF profile based on your video codec and resolution. For more information, see Default automatic selection of MXF profiles.
   - **Copy AFD to MXF**: Ignore this setting unless you have AFD signaling set up in your output video stream. If you do, and you want that information in both your video stream and the MXF wrapper, choose **Copy from video stream**. If you want your AFD signaling only in your video stream, keep the default value **Don't copy**.
5. In the **Encoding settings** section, choose a video codec.

   Your available choices in this list depend on what you choose for **MXF profile** in the previous step of this procedure:

   - When you keep the default value, **Auto**, for **MXF profile** then you can choose any video codec that is compatible with your output container.
   - When you choose a specific profile for **MXF profile**, then your choices for **Video codec** include only the codecs that are valid with that profile.

6. Optionally, specify your output video resolution for **Resolution (w x h)**. This is the other setting that MediaConvert uses when automatically determining your MXF profile.

   When you don't specify values for **Resolution (w x h)**, your output resolution is the same as your input resolution.

7. Specify your other encoding settings as usual. For information about each setting, choose the **Info** link next to the setting label.

To set up your transcoding job with an MXF output (API, CLI, or SDK)

If you use the API, CLI, or an SDK, specify the relevant settings in your JSON job specification and then submit it programmatically with your job. For more information about submitting your job programmatically, see one of the introductory topics of the *AWS Elemental MediaConvert API Reference*:

- Getting started with AWS Elemental MediaConvert using the AWS SDKs or the AWS CLI
- Getting started with AWS Elemental MediaConvert using the API

- Use the MediaConvert console to generate your JSON job specification. We recommend this approach, because the console functions as an interactive validator against the MediaConvert job schema. Follow these steps to generate your JSON job specification using the console:

  a. Follow the previous procedure for the console.

  b. In the **Job** pane on the left, under **Job settings**, choose **Show job JSON**.

Find additional information, including where each setting belongs in the job settings structure, in the *AWS Elemental MediaConvert API Reference*. Links in this list go to information about the setting in that document:

Output settings

- **Container** (`container`)
- **MXF profile** (`mxfSettings, profile`)

  For automatic profile selection, omit this setting from your JSON job specification.

- **Copy AFD to MXF** (`mxfSettings, afdSignaling`)

Encoding settings

- **Video codec** (`codec`)

  This setting is a child of `outputs, videoDescription, codecSettings`.
- **Resolution, w** (`width`)

  This setting is a child of `outputs, videoDescription`.
- **Resolution, h** (`height`)

  This setting is a child of `outputs, videoDescription`.

In the MediaConvert console, **Auto** is the default value for **MXF profile**. When you set up your job by directly editing your JSON job specification, for automatic profile selection, don't include `profile` under `MxfSettings`.

Codecs supported with each MXF profile

This table shows the codecs that AWS Elemental MediaConvert supports with each MXF profile.

MXF profile	Supported video codecs
Generic OP1a	VC3 (SD, HD) AVC Intra (SD, HD) AVC (H.264) (SD, HD, 4K) MPEG-2 (all resolutions)
D10 (SMPTE-386)	MPEG-2 (SD)
Sony XDCAM	MPEG-2 (HD)
Sony XAVC (RDD32)	XAVC (HD, 4K)
XDCAM RDD9	MPEG-2 (HD)

XDCAM RDD9 requirements

MediaConvert supports the following combinations of encoding settings when your output **MXF profile** is **XDCAM RDD9**.

In this table, read down the rows to find the **Resolution** that you want. Then read across to find a valid combination of **Bitrate**, **Frame rate**, **Interlace mode**, **GOP size**, and **Codec profile**.

Resolution	Bitrate(s)	Frame rate(s)	Interlace mode	GOP size	Codec profile
1280x720	25M 35M 50M	23.976 50 59.94	Progressive	12	Main (HD420)
1280x720	50M	23.976 25 50 59.94	Progressive	12	HD422
1280x720	50M	29.97	Progressive	15	HD422
1440x1080	17.5M 25M 35M	23.976 25	Progressive	12	Main (HD420)
1440x1080	17.5M 25M 35M	29.97	Progressive	15	Main (HD420)
1440x1080	17.5M 25M	25	Interlaced	12	Main (HD420)

Resolution	Bitrate(s)	Frame rate(s)	Interlace mode	GOP size	Codec profile
	35M				
1440x1080	17.5M 25M 35M	29.97	Interlaced	15	Main (HD420)
1920x1080	50M	23.976 25	Progressive	12	HD422
1920x1080	50M	29.97	Progressive	15	HD422
1920x1080	50M	25	Interlaced	12	HD422
1920x1080	50M	29.97	Interlaced	15	HD422

For additional information about MXF RDD9 requirements, see the SMPTE RDD 9:2013 MXF interoperability specification.

Output audio requirements for each MXF profile

This table shows the requirements for how you set up your output audio tracks in your AWS Elemental MediaConvert job. The requirements are different for each MXF profile that you choose.

> **Note**
>
> If you don't include any audio tracks in your output, then you must make sure that there is no audio at all in your job settings.
>
> - In your JSON job specification, this means removing the `AudioDescriptions` object entirely.
> - In the MediaConvert console, this means that you must delete the **Audio 1** tab that MediaConvert inserts for you by default. Delete it by choosing it, and then choosing **Remove audio** in the upper right of the **Encoding settings** section.

MXF profile	Output audio requirements
Generic OP1a	Audio codec: Uncompressed WAV
Sony XAVC (RDD32)	Audio codec: Uncompressed WAV Number of channels per track 1 Number of tracks per output: Any even number from 2 through 16 Bit depth: 24 bits Sample rate: 48 kHz
Sony XDCAM	Audio codec: Uncompressed WAV Number of channels per track: 1 Number of tracks per output: Any even number from 2 through 16, or no audio at all Bit depth: 16 bits or 24 bits; you must choose the same value for all audio tracks in the output Sample rate: 48 kHz
XDCAM RDD9	Audio codec: Uncompressed WAV Number of channels per track: 1 Number of tracks per output: 2, 4, or 8 Bit depth: 16 bits or 24 bits; you must choose the same value for all audio tracks in the output Sample rate: 48 kHz
D10 (SMPTE-386)	Audio codec: Uncompressed WAV

MXF profile	Output audio requirements
	Number of channels per track: 4 or 8 Number of tracks per output: 1, or no audio at all Bit depth: 16 bits or 24 bits; you must choose the same value for all audio tracks in the output Sample rate: 48 kHz

Default automatic selection of MXF profiles

When you choose MXF as your output video container and keep **MXF profile** out of your job settings, MediaConvert automatically chooses your MXF profile for you and creates your output accordingly. How MediaConvert makes the MXF profile selection depends on your output video codec.

> **Note**
>
> MediaConvert supports more MXF profiles through automatic profile selection than it does with manual selection.

AVC (H.264), AVC Intra, and VC3

For AVC (H.264), AVC Intra, and VC3, MediaConvert selects the Generic OP1a profile, without regard for your output encoding settings.

XAVC

For XAVC, MediaConvert selects the XAVC profile, without regard for your output encode settings.

MPEG-2

For MPEG-2, MediaConvert chooses SMPTE-386 D10 when your output encoding characteristics conform to the D10 specification. Otherwise, MediaConvert selects the XDCAM RDD9 profile. The following table details the relevant encoding settings and the values that you must set to have MediaConvert automatically select the D10 profile.

> **Note**
>
> For your output resolution and frame rate, you can specify an allowed value explicitly with the settings **Resolution** and **Frame rate**, or you can choose the value **Follow source** and use an input that has an allowed value.

Setting	Allowed values for D10 profile
Combination of: **Resolution** (wxh), **Frame rate**, **Interlace mode**	Allowed combinations: • 720x512, 29.97, interlaced • 720x486, 29.97, interlaced • 720x486, 29.97, interlaced • 720x608, 25, interlaced • 720x576, 25, interlaced
GOP size	1 frame
Profile, for your video codec	4:2:2
Syntax This setting is visible on the MediaConv ert console when you choose MXF for your container, MPEG-2 for your video codec, and 4:2:2 for your codec profile.	D10
Number of audio tracks Each output track is represented on the MediaConvert console as one **Audio** tab and in your JSON job specification as a direct child of `AudioDescriptions` .	• 0 (no audio) • 1

Setting	Allowed values for D10 profile
Captions type You specify this on the MediaConvert console with the setting **Destination type**.	• Burn in • Embedded

This excerpt from a JSON job specification shows the same settings as you would submit them programmatically:

```
{
  "height": 512,
  "width": 720,
  "codecSettings": {
    "codec": "MPEG2",
    "mpeg2Settings": {
      "bitrate": 30000000,
      "syntax": "D_10",
      "framerateDenominator": 1001,
      "framerateControl": "SPECIFIED",
      "framerateNumerator": 30000,
      "numberBFramesBetweenReferenceFrames": 0,
      "gopSize": 1.0,
      "gopSizeUnits": "FRAMES",
      "codecLevel": "MAIN",
      "codecProfile": "PROFILE_422",
      "rateControlMode": "CBR",
      "interlaceMode": "TOP_FIELD"
    }
  }
}
```

Video passthrough

MediaConvert supports video passthrough for I-frame only inputs and **MXF** or **QuickTime** outputs. Use video passthrough to repackage one or more inputs without any video encoding.

Supported video codecs for video passthrough

MediaConvert supports video passthrough for the following I-frame only input video codecs:

- Apple ProRes
- AVC-Intra
- DV/DVCPRO
- JPEG 2000 (J2K)
- VC-3

Job settings requirements

When you use video passthrough, you can repackage your input video but **you cannot re-encode or otherwise modify the video essence**. Jobs have the following restrictions.

Inputs

Specify one or more inputs with one of the following supported video codecs: Apple ProRes, AVC-Intra, DV/DVCPRO, JPEG 2000 (J2K), and VC-3.

If you specify multiple inputs, each of your input's encoding attributes must exactly match, including video codec, frame size, profile, frame rate, and color space.

Input clips

(Optional) MediaConvert supports input clipping when you use video passthrough.

Output container

You must choose **MXF**, or **QuickTime** (MOV).

Frame rate

Keep the default setting, **Follow source**. You cannot specify a different output frame rate than your input.

Audio codec

(Optional) MediaConvert supports audio encoding when you use video passthrough.

Dolby Digital passthrough

(Optional) MediaConvert supports **Dolby Digital passthrough** when you use video passthrough.

If you specify multiple inputs, each of your input's Dolby Digital streams must have identical encoding attributes.

Captions

(Optional) MediaConvert supports sidecar formats when you use video passthrough.

Using video passthrough

The following sections describe how to configure your job settings with Video passthrough.

MediaConvert console

To specify Video passthrough using the MediaConvert console:

1. Open the Create job page in the MediaConvert console.
2. On the **Create job** page, provide transcode instructions and job settings. For more information, see Configuring jobs in MediaConvert.
3. Specify one or more supported inputs. If you specify multiple inputs, each of your input's encoding attributes must exactly match, including video codec, frame size, profile, frame rate, and color space.
4. Next to **Output groups**, select **Add** .
5. Keep **File group** selected and choose **Select**.
6. In the **Output settings**, under **Container**, choose **QuickTime** or **MXF**.
7. In **Encoding** settings, under **Video codec**, choose **Passthrough**.

API, SDK, or AWS Command Line Interface (AWS CLI)

The following is an excerpt of a job settings JSON that specifies Video passthrough for an Apple ProRes workflow with two inputs:

```
{
  "Settings": {
    "Inputs": [
      {
        "FileInput": "s3://DOC-EXAMPLE-BUCKET/prores.mov"
      },
      {
        "FileInput": "s3://DOC-EXAMPLE-BUCKET/prores-2.mov"
      }
    ],
```

```
    "OutputGroups": [
      {
        "Name": "File Group",
        "OutputGroupSettings": {
          "Type": "FILE_GROUP_SETTINGS",
          "FileGroupSettings": {
            "Destination": "s3://DOC-EXAMPLE-BUCKET/passthrough-output.mov"
          }
        },
        "Outputs": [
          {
            "VideoDescription": {
              "CodecSettings": {
                "Codec": "PASSTHROUGH"
              }
            },
            "ContainerSettings": {
              "Container": "MOV",
              "MovSettings": {}
            }
          }
        ]
      }
    ]
  }
}
```

AAC audio codec support

This topic covers AAC audio codec support and lists the supported combinations of coding mode, profile, sampling rate, and bitrate.

This section explains how to set the following four properties of the AAC audio codec:

- **Profile**
- **Coding mode**
- **Sample rate**
- **Bitrate**

The settings that you choose for **Profile**, **Coding mode**, **Sample rate**, and **Bitrate** are limited by the AAC codec. The following procedure shows how to configure a supported combination of settings.

Choosing AAC audio codec settings

1. Choose a **Profile**.
2. Choose a **Coding mode** that is valid with that profile. The **Coding mode** determines the number of audio channels and the audio channel layout metadata.
3. Choose a **Sample rate** that is valid for that combination of profile and coding mode.
4. Choose a **Bitrate** that falls within the range that is supported for that sample rate.

CBR Bitrate control mode

AWS Elemental MediaConvert supports the following combinations of AAC settings when your output **Bitrate control mode** is **CBR**.

To use this table, choose a profile from the **Profile** column. Then read across to find a valid combination of **Coding mode**, **Sample rate**, and **Bitrate**:

Profile	Coding mode(s)	Sample rate(s) (Hz)	Bitrate(s) (bit/s)
LC	1.0	8000, 12000	8000, 10000, 12000, 14000
LC	1.0	16000	8000, 10000, 12000, 14000, 16000, 20000, 24000, 28000
LC	1.0	22050, 24000	24000, 28000
LC	1.0	32000	32000, 40000, 48000, 56000, 64000, 80000, 96000, 112000, 128000, 160000, 192000
LC	1.0	44100	56000, 64000, 80000, 96000, 112000, 128000, 160000, 192000, 224000, 256000

Profile	Coding mode(s)	Sample rate(s) (Hz)	Bitrate(s) (bit/s)
LC	1.0	48000	56000, 64000, 80000, 96000, 112000, 128000, 160000, 192000, 224000, 256000, 288000
LC	1.0	88200	288000
LC	1.0	96000	128000, 160000, 192000, 224000, 256000, 288000
LC	2.0	8000, 12000	16000, 20000
LC	2.0	16000	16000, 20000, 24000, 28000, 32000
LC	2.0	22050, 24000	32000
LC	2.0	32000	40000, 48000, 56000, 64000, 80000, 96000, 112000, 128000, 160000, 192000, 224000, 256000, 288000, 320000, 384000
LC	2.0	44100	64000, 80000, 96000, 112000, 128000, 160000, 192000, 224000, 256000, 288000, 320000, 384000, 448000, 512000

Profile	Coding mode(s)	Sample rate(s) (Hz)	Bitrate(s) (bit/s)
LC	2.0	48000	64000, 80000, 96000, 112000, 128000, 160000, 192000, 224000, 256000, 288000, 320000, 384000, 448000, 512000, 576000
LC	2.0	88200	576000
LC	2.0	96000	256000, 288000, 320000, 384000, 448000, 512000, 576000
LC	5.1	32000	160000, 192000, 224000, 256000, 288000, 320000, 384000, 448000, 512000, 576000, 640000, 768000
LC	5.1	44100	256000, 288000, 320000, 384000, 448000, 512000, 576000, 640000
LC	5.1	48000	256000, 288000, 320000, 384000, 448000, 512000, 576000, 640000, 768000
LC	5.1	96000	640000, 768000
HEV1	1.0	22050, 24000	8000, 10000

Profile	Coding mode(s)	Sample rate(s) (Hz)	Bitrate(s) (bit/s)
HEV1	1.0	32000	12000, 14000, 16000, 20000, 24000, 28000, 32000, 40000, 48000, 56000, 64000
HEV1	1.0	44100, 48000	20000, 24000, 28000, 32000, 40000, 48000, 56000, 64000
HEV1	2.0	32000	16000, 20000, 24000, 28000, 32000, 40000, 48000, 56000, 64000, 80000, 96000, 112000, 128000
HEV1	2.0	44100	16000, 20000, 24000, 28000, 32000, 40000, 48000, 56000, 64000, 80000, 96000
HEV1	2.0	48000	16000, 20000, 24000, 28000, 32000, 40000, 48000, 56000, 64000, 80000, 96000, 112000, 128000
HEV1	2.0	96000	96000, 112000, 128000
HEV1	5.1	32000	64000, 80000, 96000, 112000, 128000, 160000, 192000, 224000, 256000, 288000, 320000

Profile	Coding mode(s)	Sample rate(s) (Hz)	Bitrate(s) (bit/s)
HEV1	5.1	44100	64000, 80000, 96000, 112000, 128000, 160000, 192000, 224000
HEV1	5.1	48000	64000, 80000, 96000, 112000, 128000, 160000, 192000, 224000, 256000, 288000, 320000
HEV1	5.1	96000	256000, 288000, 320000
HEV2	2.0	22050, 24000	8000, 10000
HEV2	2.0	32000	12000, 14000, 16000, 20000, 24000, 28000, 32000, 40000, 48000, 56000, 64000
HEV2	2.0	44100, 48000	20000, 24000, 28000, 32000, 40000, 48000, 56000, 64000

VBR Bitrate control mode

When your output **Bitrate control mode** is **VBR**, AWS Elemental MediaConvert creates AAC outputs with the following approximate bitrates:

Profile	Coding mode(s)	Bitrate quality	Approximate bitrate per channel (bits/s)
LC	1.0	Low	52000
LC	2.0, 5.1	Low	48000

Profile	Coding mode(s)	Bitrate quality	Approximate bitrate per channel (bits/s)
LC	1.0	Medium low	68000
LC	2.0, 5.1	Medium low	64000
LC	1.0	Medium high	100000
LC	2.0, 5.1	Medium high	96000
LC	1.0	High	132000
LC	2.0, 5.1	High	128000

Using MediaConvert to create outputs with only audio

You can use AWS Elemental MediaConvert to create outputs that contain only audio, without video. With audio-only outputs, MediaConvert supports a more limited number of codec and container combinations for input and output files.

> **Note**
>
> The restrictions and procedures in this chapter apply to outputs that don't have video in the container. This includes the following:
>
> - Outputs in **File** output groups that don't have video included
> - Streaming **HLS** output groups that contain only audio outputs
> - Streaming **DASH** output groups that contain only audio outputs
>
> When you set up streaming output packages that contain audio, video, and captions, you create separate outputs for each element inside the output package. These are not audio-only outputs as described in this chapter. For more information about setting up streaming outputs, see Creating outputs in ABR streaming output groups.

Topics

- Setting up audio-only outputs
- Supported codecs and containers for audio-only outputs
- Feature limitations

Setting up audio-only outputs

You set up an audio-only output in the same way that you set up an output that contains video, except that you don't include video or captions.

MediaConvert generates the following files for audio-only outputs:

- **File** output groups: One separate audio-only file for each output.
- **HLS** output groups: A single rendition in the ABR stack for each output.
- **DASH ISO** output groups: A single rendition in the ABR stack for each output.

> **Note**
>
> For AAC streaming outputs, the initial segment is longer in duration than the others. This is because, with AAC, the initial segment must contain silent AAC pre-roll samples before the audible part of the segment. MediaConvert accounts for these extra samples in the timestamps, so the audio plays back correctly.

To create an audio-only output (console)

1. To confirm that MediaConvert supports your input files, consult the input table in Supported codecs and containers for audio-only outputs.
2. Set up your job as described in Configuring jobs in MediaConvert, but with the following differences:
   - Remove the **Video** tab in the **Encoding settings** section of your output.
   - Choose a supported output container and audio codec from the output table in Supported codecs and containers for audio-only outputs.
   - Include only one group of audio settings per output. That is, don't choose **Add audio** to create an **Audio 2** tab under **Encoding settings**.
3. If your outputs are in an **HLS** output group, choose a container for the audio-only output.

Under **Output settings**, in the **Audio** section, for **Container for audio-only output**, choose **MPEG-2 Transport Stream** to create a file in an MPEG2-TS container. Keep the default value **Automatic** to create a raw audio-only file with no container.

To create an audio-only output (API, SDK, and AWS CLI)

1. To confirm that MediaConvert supports your input files, consult the input table in Supported codecs and containers for audio-only outputs.
2. Set up your JSON job specification. Either manually edit your JSON file, or use the console to generate it as follows:

   a. Follow the previous procedure for the console.

   b. In the **Job** pane on the left, under **Job settings**, choose **Show job JSON**.

   The JSON job specification for audio-only jobs differs from standard jobs as follows:

   - Exclude each instance of `VideoDescription` and its children from the `Outputs` portion of your job JSON.
   - For each output, include only one child group of audio settings under `AudioDescriptions`.
   - For audio-only outputs in an **HLS** output group, specify a container for the audio-only output. Under `Outputs`, `OutputSettings`, `HlsSettings` include the property `HlsAudioOnlyContainer`. Set it to `M2TS` to create a file in an MPEG2-TS container. Set it to `AUTOMATIC` to create a raw audio-only file with no container. `AUTOMATIC` is the default behavior.

Supported codecs and containers for audio-only outputs

Input codecs and containers supported for audio-only

MediaConvert supports the following combinations of input container and codec for audio-only outputs.

Container	Supported audio Codecs

Container	Supported audio Codecs
ASF, WMA, WMV (Advanced Systems Format)	WMA WMA2 WMA Pro
FLAC (Free Lossless Audio Codec)	FLAC
Matroska	FLAC Opus
MOV (Apple QuickTime)	PCM
MP3 (MPEG-1 Layer 3)	MP3
MP4 (MPEG-4)	AAC
MPEG-TS (MPEG-2 Transport Stream)	MPEG-2 PCM
OGG, OGA	FLAC Opus Vorbis
WAV	PCM

Container	Supported audio Codecs
No container	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3) FLAC PCM

Output codecs and containers supported for audio-only

MediaConvert supports the following combinations of output container and codec for creating audio-only outputs.

Container	Audio Codecs
DASH (Dynamic Adaptive Streaming over HTTP)	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3) Dolby Digital Plus JOC (Atmos)
HLS (Apple HTTP Live Streaming)	AAC Dolby Digital (AC3)
MP4 (MPEG-4)	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3)
MPEG-TS (MPEG-2 Transport Stream)	

Container	Audio Codecs
	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3) MPEG-2
No container	AAC AIFF Dolby Digital (AC3) Dolby Digital Plus (EAC3) FLAC MPEG-2 MP3 WAV

Feature limitations

In any job that contains audio-only outputs, you can't use these features:

- Job progress status update

In any audio-only output, you can't use these features:

- Captions
- Still or motion graphic overlay (image inserter, motion image inserter)

Supported containers and codecs reference tables

The following sections provide reference tables for containers and codecs that MediaConvert supports.

To use these tables, first find the container or codec using the links following this introduction. For containers, the tables provide details for which video or audio codecs are supported, and whether they're supported for inputs or outputs. For codecs, the tables provide details for which containers are supported, for inputs and outputs as well. Note that inputs or outputs labeled like *Audio-only input* or *Audio-only output* refer to support for audio-only workflows. For more information, see Audio-only outputs.

For quick reference tables that show which input formats MediaConvert supports, see Supported input formats.

For quick reference tables that show which output formats MediaConvert supports, see Supported output formats.

Containers

3G2 | 3GP | ASF | AVI | CMAF | DASH | F4V | FLV | GIF | HLS | IMF | Matroska | MOV | MP3 | MP4 | MPEG-1 | MPEG-PS | MPEG-TS | MSS | MXF | OGG | WAV | WebM | WMA | WMV | Y4M | No container

Codecs

AAC | AIFF | AMR-NB | AMR-WB | Apple ProRes | AV1 | AVC (H.264) | AVC-Intra | Canopus HQ | Dolby Atmos | Dolby Digital (AC3) | Dolby Digital Plus (EAC3) | Dolby E | DV/DVCPRO | DV25 | DV50 | DVCPro HD | DivX/Xvid | FLAC | GIF | H.261 | H.262 | H.263 | HEVC (H.265) | J2K | MJPEG | MP3 | MPEG-1 | MPEG-2 | MPEG-4 Part 2 | MPEG Audio | Opus | Panasonic P2 | PCM | QuickTime RLE | Sony XDCAM | Sony XDCAM MPEG-4 Proxy | VC-1 | VC-3 | Vorbis | VP8 | VP9 | WMA | WMA2 | WMA Pro | XAVC

Topics

- Supported containers
- Supported codecs

Supported containers

This section contains reference tables for input and output containers that MediaConvert supports. The tables show the container, support on the input or output side, and video or audio codec support in the container. For more information about the codec, or to see codec support in other containers, choose the codec link.

3G2, 3GP

Container	Input / Output	Supported video codec	Supported audio codec
3G2 3GP	Input	AVC (H.264) H.263 MPEG-4 part 2	AAC AMR-NB AMR-WB
	Output	*Not supported*	*Not supported*

ASF, WMV, WMA (Advanced Systems Format)

Container	Input / Output	Supported video codec	Supported audio codec
ASF WMV	Input	VC-1	WMA WMA2 WMA Pro
	Output	*Not supported*	*Not supported*
WMA	Audio-only input	*Not applicable*	WMA WMA2 WMA Pro

Container	Input / Output	Supported video codec	Supported audio codec
	Audio-only output	*Not applicable*	*Not supported*

AVI (Audio Video Interleave)

Container	Input / Output	Supported video codec	Supported audio codec
AVI	Input	Canopus HQ DivX/Xvid DV/DVCPRO MJPEG Uncompressed	Dolby Digital (AC3) Dolby Digital Plus (EAC3) Dolby E MP3 MPEG Audio PCM
	Output	*Not supported*	*Not supported*

CMAF (Common Media Application Format)

Container	Input / Output	Supported video codec	Supported audio codec
CMAF DASH	Input	*Not supported*	*Not supported*
	Output	AV1 AVC (H.264) HEVC (H.265)	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3)

Container	Input / Output	Supported video codec	Supported audio codec
CMAF HLS	Input	*Not supported*	*Not supported*
	Output	AV1 AVC (H.264) HEVC (H.265)	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3)

Note

For HLS output codec recommendations from Apple, see: https://developer.apple.com/documentation/http-live-streaming/hls-authoring-specification-for-apple-devices.

DASH (Dynamic Adaptive Streaming over HTTP)

Container	Input / Output	Supported video codec	Supported audio codec
DASH	Input	*Not supported*	*Not supported*
	Output	AV1 AVC (H.264) HEVC (H.265) VP8 VP9	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3)
	Audio-only input	*Not supported*	*Not supported*
	Audio-only output	*Not applicable*	AAC Dolby Digital (AC3)

Container	Input / Output	Supported video codec	Supported audio codec
			Dolby Digital Plus (EAC3) Dolby Digital Plus JOC (Atmos)

For information about CMAF DASH, see CMAF containers.

FLV, F4V (MPEG-4 Flash)

Container	Input / Output	Supported video codec	Supported audio codec
FLV F4V	Input	AVC (H.264) H.263	AAC
F4V	Output	AVC (H.264) MPEG-2	AAC

GIF

Container	Input / Output	Supported video codec
GIF	Input	GIF
	Output	*Not supported*

HLS (Apple HTTP Live Streaming)

Container	Input / Output	Supported video codec	Supported audio codec
HLS	Input	AVC (H.264)	AAC

Container	Input / Output	Supported video codec	Supported audio codec
		HEVC (H.265) MPEG-2	Dolby Digital (AC3) Dolby Digital Plus (EAC3)
	Output	AVC (H.264) HEVC (H.265)	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3)
	Audio-only input	*Not applicable*	*Not supported*
	Audio-only output	*Not applicable*	AAC Dolby Digital (AC3)

For more information about HLS inputs and input requirements, see Using HLS inputs with MediaConvert.

For information about CMAF HLS, see CMAF containers.

Note

When outputting HEVC in an HLS container, we recommend using a CMAF output group for the widest player compatibility. For more details, see https://developer.apple.com/documentation/http-live-streaming/hls-authoring-specification-for-apple-devices.

IMF (Interoperable Master Format)

Container	Input / Output	Supported video codec	Supported audio codec
IMF	Input	Apple ProRes	PCM

Container	Input / Output	Supported video codec	Supported audio codec
		JPEG 2000 (J2K)	
	Output	*Not supported*	*Not supported*

Specify your input IMF by providing the path to your Composition Playlist (CPL). If the CPL is in an incomplete IMP, also specify any supplemental IMPs.

For information about using IMF inputs to create Dolby Vision outputs, see Dolby Vision input format support and job setting requirements.

Matroska

Container	Input / Output	Supported video codec	Supported audio codec
Matroska	Input	AVC (H.264) MPEG-2 MPEG-4 part 2 VC-1	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3) FLAC Opus PCM WMA WMA2
	Output	*Not supported*	*Not supported*
	Audio-only input	*Not applicable*	FLAC OPUS

Container	Input / Output	Supported video codec	Supported audio codec
	Audio-only output	*Not applicable*	*Not supported*

MOV (Apple QuickTime)

Container	Input / Output	Supported video codec	Supported audio codec
MOV	Input	Apple ProRes AVC (H.264) AVC-Intra DivX/Xvid DV/DVCPRO H.261 H.262 H.263 HEVC (H.265) JPEG 2000 (J2K) MJPEG MPEG-2 MPEG-4 part 2 QuickTime RLE Uncompressed	AAC MP3 PCM
	Output	AVC (H.264)	AAC

Container	Input / Output	Supported video codec	Supported audio codec
		MPEG-2	Dolby Digital (AC3) Dolby Digital Plus (EAC3) WAV
		Apple ProRes	AIFF
	Audio-only input	*Not applicable*	PCM
	Audio-only output	*Not applicable*	*Not supported*

MOV inputs must be self-contained. References to external files are not supported.

For information about using MOV inputs to create Dolby Vision outputs, see Dolby Vision input format support and job setting requirements.

MP3 (MPEG-1 Layer 3)

Container	Input / Output	Supported audio codec
MP3	Audio-only input	MP3
	Audio-only output	MP3

MediaConvert does not read ID3 metadata or tags from MP3 inputs.

MP4 (MPEG-4)

Container	Input / Output	Supported video codec	Supported audio codec
MP4	Input	AV1 AVC (H.264)	AAC Dolby Digital (AC3)

Container	Input / Output	Supported video codec	Supported audio codec
		AVC-Intra DivX/Xvid H.261 H.262 H.263 HEVC (H.265) JPEG 2000 (J2K) MPEG-2 MPEG-4 part 2 VC-1 VP9 Uncompressed	Dolby Digital Plus (EAC3) PCM WMA WMA2
	Output	AV1 AVC (H.264) HEVC (H.265)	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3)
	Audio-only input	*Not applicable*	AAC
	Audio-only output	*Not applicable*	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3)

MediaConvert doesn't support fragmented MP4 initialization segments. Your MP4 input must be self-contained.

MPEG-1 (MPEG-1 System Stream)

Container	Input / Output	Supported video codec	Supported audio codec
MPEG-1	Input	MPEG-1 MPEG-2	AAC AIFF Dolby Digital (AC3) Dolby Digital Plus (EAC3) MPEG Audio PCM
	Output	*Not supported*	*Not supported*

MPEG-PS (MPEG Program Stream)

Container	Input / Output	Supported video codec	Supported audio codec
MPEG-PS	Input	MPEG-2	MPEG audio
	Output	*Not supported*	*Not supported*

MPEG-TS (MPEG-2 Transport Stream)

Container	Input / Output	Supported video codec	Supported audio codec
MPEG-TS	Input	AVC (H.264)	AAC

Container	Input / Output	Supported video codec	Supported audio codec
		HEVC (H.265) MPEG-2 VC-1	AIFF Dolby Digital (AC3) Dolby Digital Plus (EAC3) Dolby E MPEG Audio PCM WMA WMA2
	Output	AVC (H.264) HEVC (H.265) MPEG-2	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3) MPEG-2 PCM/WAV
	Audio-only input	*Not applicable*	MPEG-2 PCM

Container	Input / Output	Supported video codec	Supported audio codec
	Audio-only output	*Not applicable*	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3) MPEG-2

MSS (Microsoft Smooth Streaming)

Container	Input / Output	Supported video codec	Supported audio codec
MSS	Input	*Not supported*	*Not supported*
	Output	AVC (H.264)	AAC Dolby Digital (AC3) Dolby Digital Plus (EAC3)

MXF (Material Exchange Format)

Container	Input / Output	Supported video codec	MXF Profile	Supported audio codec
MXF	Input	Apple ProRes AVC (H.264) AVC-Intra DV/DVCPRO	*Not applicable*	AAC AIFF Dolby E MPEG Audio

Container	Input / Output	Supported video codec	MXF Profile	Supported audio codec
		DV25 DV50 DVCPro HD JPEG 2000 (J2K) MPEG-2 Panasonic P2 Sony XDCam Sony XDCam MPEG-4 Proxy Uncompressed		PCM
	Output	AVC (H.264) AVC-Intra MPEG-2 VC-3	Generic OP1a	PCM/WAV
		MPEG-2	XDCAM RDD9	
		MPEG-2	D10 (SMPTE-386)	
		XAVC	Sony XAVC (RDD32)	

MXF inputs must be self-contained. References to external files are not supported.

For more information about creating MXF outputs, see Creating MXF outputs with AWS Elemental MediaConvert.

For information about using MXF inputs to create Dolby Vision outputs, see Dolby Vision input format support and job setting requirements.

OGG (Ogg Vorbis Audio)

Container	Input / Output	Supported audio codec
OGG	Audio-only input	Opus Vorbis FLAC
	Audio-only output	*Not supported*

WAV (Waveform Audio File Format)

Container	Input / Output	Supported audio codec
WAV	Audio-only input	PCM
	Audio-only output	PCM

WebM

Container	Input / Output	Supported video codec	Supported audio codec
WebM	Input	VP8 VP9	Opus Vorbis
	Output	VP8 VP9	Opus Vorbis

Y4M

Container	Input / Output	Supported video codec	Supported audio codec
Y4M	Input	*Not supported*	*Not supported*
	Output	Uncompressed	*Not supported*

MediaConvert supports uncompressed Y4M outputs with I420, I422, or I444 four character codes (FOURCCs).

No container

Container	Input / Output	Supported video codec	Supported audio codec
No container	Video-only input	DV/DVCPRO AVC (H.264) HEVC (H.265) MPEG-1 MPEG-2	*Not applicable*
	Video-only output	AVC-Intra AVC (H.264) HEVC (H.265) MPEG-2 VC-3 XAVC	*Not applicable*
	Audio-only input	*Not applicable*	AAC

Container	Input / Output	Supported video codec	Supported audio codec
			Dolby Digital (AC3) Dolby Digital Plus (EAC3) FLAC PCM
	Audio-only output	*Not applicable*	AAC AIFF Dolby Digital (AC3) Dolby Digital Plus (EAC3) FLAC MPEG-2 MP3 PCM

Supported codecs

This section contains reference tables for input and output codecs that MediaConvert supports. The tables show the codec, support on the input or output side, and container support for the codec. For more information about the container, or to see container support for other codecs, choose the container link.

AAC (Advanced Audio Codec)

Audio codec	Input / Output	Supported container
AAC	Input	3G2 3GP HLS MPEG-4 Flash Matroska MOV MP4 MPEG-1 MPEG-TS MXF
	Output	CMAF DASH CMAF HLS DASH HLS MOV MP4 MPEG-4 Flash MPEG-TS MSS No container

Audio codec	Input / Output	Supported container
	Audio-only input	MP4 No container
	Audio-only output	DASH HLS MPEG-TS MP4 No container

For information about what output AAC profiles, coding modes, sample rates , and bitrates MediaConvert supports, see AAC audio codec support.

AIFF

Audio codec	Input / Output	Supported container
AIFF	Input	MPEG-TS MPEG-1 MXF
	Output	MOV No container
	Audio-only input	*Not supported*
	Audio-only output	No container

AMR-NB, AMR-WB

Audio codec	Input / Output	Supported container
AMR-NB AMR-WB	Input	3G2 3GP
	Output	*Not supported*

Apple ProRes

Video codec	Input / Output	Supported container	Supported formats
Apple ProRes	Input	IMF MOV MXF	Apple ProRes 4444 XQ Apple ProRes 4444 Apple ProRes 422 HQ Apple ProRes 422 Apple ProRes LT Apple ProRes Proxy
	Output	MOV	

For more information about Apple ProRes formats, see https://support.apple.com/en-us/HT202410.

Note

To preserve 4:4:4 chroma subsampling in your Apple ProRes outputs:

- You cannot include any of the following Preprocessors: **Dolby Vision**, **HDR10+**, or **Noise reducer**.

- You must use the Duplicate Drop as the frame rate conversion algorithm (when using frame rate conversion).
- You cannot mix RGB and non RGB inputs.
- You cannot mix 4:4:4 inputs with other non-4:4:4 inputs.
- You can only use the NexGuard File Maker preprocessor.

AV1

Video codec	Input / Output	Supported container
AV1	Input	MP4
	Output	CMAF DASH CMAF HLS DASH MP4

Note

For HLS output codec recommendations from Apple, see https://developer.apple.com/documentation/http-live-streaming/hls-authoring-specification-for-apple-devices.

AVC (H.264)

Video codec	Input / Output	Supported container
AVC (H.264)	Input	3G2 3GP MPEG-4 Flash HLS

Video codec	Input / Output	Supported container
		Matroska MOV MP4 MPEG-TS MXF No container
	Output	CMAF DASH CMAF HLS DASH F4V HLS MOV MP4 MPEG-TS MSS MXF No container

AVC-Intra

Video codec	Input / Output	Supported container	Supported formats
AVC-Intra	Input	MOV	AVC-Intra 50

Video codec	Input / Output	Supported container	Supported formats
		MP4 MXF	AVC-Intra 100 AVC-Intra 200 AVC-Intra 2K4:2:2 AVC-Intra 4K4:2:2
	Output	MXF No container	AVC-Intra 50 AVC-Intra 100 AVC-Intra 200 AVC-Intra 2K4:2:2 AVC-Intra 4K4:2:2

MediaConvert only supports YUV AVC-Intra inputs, it does not support RGB AVC-Intra inputs.

Canopus HQ

Video codec	Input / Output	Supported container
Canopus HQ	Input	AVI
	Output	*Not supported*

Dolby Digital (AC3)

Audio codec	Input / Output	Supported container
Dolby Digital (AC3)	Input	AVI HLS Matroska

Audio codec	Input / Output	Supported container
		MP4 MPEG-1 MPEG-TS
	Output	CMAF DASH CMAF HLS DASH HLS MOV MP4 MPEG-TS MSS No container
	Audio-only input	No container
	Audio-only output	DASH HLS MP4 MPEG-TS No container

Dolby Digital Plus (EAC3)

Audio codec	Input / Output	Supported container
Dolby Digital Plus (EAC3)	Input	AVI HLS Matroska MP4 MPEG-1 MPEG-TS
	Output	CMAF DASH CMAF HLS HLS MOV MP4 MPEG-TS MSS No container
	Audio-only input	No container
	Audio-only output	DASH MP4 MPEG-TS No container

Dolby Digital Plus JOC (Atmos)

For more information, see Creating Dolby Atmos outputs with AWS Elemental MediaConvert.

Dolby E

Audio codec	Input / Output	Supported container
Dolby E	Input	AVI MPEG-TS MXF No container
	Output	*Not supported*

DV/DVCPRO

Video codec	Input / Output	Supported container
DV/DVCPRO	Input	AVI MOV MXF No container
	Output	*Not supported*

DV25, DV50

Video codec	Input / Output	Supported container
DV25 DV50	Input	MXF

Video codec	Input / Output	Supported container
	Output	*Not supported*

DVCPro HD

Video codec	Input / Output	Supported container
DVCPro HD	Input	MXF
	Output	*Not supported*

DivX/Xvid

Video codec	Input / Output	Supported container
DivX/Xvid	Input	AVI MOV MP4
	Output	*Not supported*

FLAC

Audio codec	Input / Output	Supported container
FLAC	Input	Matroska
	Output	*Not supported*
	Audio-only input	Matroska OGA OGG

Audio codec	Input / Output	Supported container
		No container
	Audio-only output	No container

GIF

Video codec	Input / Output	Supported container
GIF	Input	GIF
	Output	*Not supported*

H.261

Video codec	Input / Output	Supported container
H.261	Input	MOV MP4
	Output	*Not supported*

H.262

Video codec	Input / Output	Supported container
H.262	Input	MOV MP4
	Output	*Not supported*

H.263

Video codec	Input / Output	Supported container
H.263	Input	3G2 3GP MPEG-4 Flash MOV MP4
	Output	*Not supported*

HEVC (H.265)

Video codec	Input / Output	Supported container
HEVC (H.265)	Input	HLS MOV MP4 MPEG-TS No container
	Output	CMAF HLS CMAF DASH DASH HLS MP4 MPEG-TS

Video codec	Input / Output	Supported container
		No container

Note

When outputting HEVC in an HLS container, we recommend using a CMAF output group for the widest player compatibility. For more details, see https://developer.apple.com/documentation/http-live-streaming/hls-authoring-specification-for-apple-devices.

JPEG 2000 (J2K)

Video codec	Input / Output	Supported container
JPEG 2000 (J2K)	Input	IMF MOV MP4 MXF
	Output	*Not supported*

MJPEG (Motion JPEG)

Video codec	Input / Output	Supported container
MJPEG	Input	AVI MOV
	Output	*Not supported*

MP3 (MPEG-1 Layer 3)

Audio codec	Input / Output	Supported container
MP3	Input	AVI MOV
	Output	MOV
	Audio-only input	MP3
	Audio-only output	No container

MPEG-1

Video codec	Input / Output	Supported container
MPEG-1	Input	MPEG-1 No container
	Output	*Not supported*

MPEG-2 (MPEG-1 Layer II)

Video codec	Input / Output	Supported container
MPEG-2 Video	Input	HLS Matroska MOV MP4 MPEG-1 MPEG-PS

Video codec	Input / Output	Supported container
		MPEG-TS MXF No container
	Output	MOV MPEG-4 Flash MPEG-TS MXF No container

Audio codec	Input / Output	Supported container
MPEG-2 Audio	Audio-only input	MPEG-TS
	Audio-only output	MPEG-TS No container

MPEG-4 Part 2

Video codec	Input / Output	Supported container
MPEG-4 Part 2	Input	3G2 3GP Matroska MOV MP4

Video codec	Input / Output	Supported container
	Output	*Not supported*

MPEG Audio

Audio codec	Input / Output	Supported container
MPEG Audio	Input	AVI MPEG-1 MPEG-PS MPEG-TS MXF
	Output	*Not supported*

Opus, Vorbis

Audio codec	Input / Output	Supported container
Opus Vorbis	Input	Matroska OGA OGG WebM
	Output	WebM
	Audio-only input	Matroska OGA OGG

Audio codec	Input / Output	Supported container
	Audio-only output	*Not supported*

Panasonic P2

Video codec	Input / Output	Supported container
Panasonic P2	Input	MXF
	Output	*Not supported*

PCM

Audio codec	Input / Output	Supported container
PCM	Input	AVI IMF Matroska MOV MP4 MPEG-1 MPEG-TS MXF No container
	Output	MOV MPEG-TS MXF

Audio codec	Input / Output	Supported container
		No container
	Audio-only input	MOV MPEG-TS WAV
	Audio-only output	WAV No container

Quicktime RLE (Quicktime Animation)

Video codec	Input / Output	Supported container
Quicktime RLE	Input	MOV
	Output	*Not supported*

Sony XDCAM

Video format	Input / Output	Supported container
Sony XDCAM	Input	MXF
	Output	MXF

Sony XDCAM MPEG-4 Proxy

Video format	Input / Output	Supported container
Sony XDCAM	Input	MXF
	Output	*Not supported*

VC-1

Video codec	Input / Output	Supported container
VC-1	Input	ASF Matroska MP4 MPEG-TS WMV
	Output	*Not supported*

VC-3

Video codec	Input / Output	Supported container
VC-3	Input	MXF
	Output	MXF No container

VP8

Video codec	Input / Output	Supported container
VP8	Input	WebM
	Output	DASH WebM

VP9

Video codec	Input / Output	Supported container
VP9	Input	MP4 WebM
	Output	DASH WebM

WMA, WMA2

Audio codec	Input / Output	Supported container
WMA WMA2	Input	ASF Matroska MP4 MPEG-TS WMV
	Output	*Not supported*
	Audio-only input	ASF WMA WMV
	Audio-only output	*Not supported*

WMA Pro

Video codec	Input / Output	Supported container
WMA Pro	Input	ASF WMV
	Output	*Not supported*
	Audio-only input	ASF WMA WMV
	Audio-only output	*Not supported*

XAVC

Format	Input / Output	Supported container	Supported XAVC profiles
XAVC	Output	MXF No container	XAVC HD XAVC HD Intra CBG XAVC 4K XAVC 4K Intra CBG XAVC 4K Intra VBR

For more information about the XAVC format, see: https://pro.sony/ue_US/technology/xavc.

XAVC inputs are supported, as they are a subset of MXF containers with AVC (H.264) video codecs.

Captions supported by AWS Elemental MediaConvert

MediaConvert supports a wide range of captions conversion workflows. But you can't convert every supported input captions format to every supported output captions format. Each video container supports a subset of all possible captions types. Furthermore, MediaConvert supports a subset of all transformations theoretically allowed within the input and output video container type.

For information about how to set up captions in your input, see Setting up captions in AWS Elemental MediaConvert jobs.

For information about how to set up captions in your output, see Setting up captions in outputs.

To look up whether MediaConvert supports your captions workflow, choose the topic from the following list that corresponds to your input captions. Find the table that corresponds to your video output container. Then find the row for your input captions. Check the supported output captions format column to see the available output captions formats.

- If your primary MediaConvert input contains your captions, see Supported captions workflows, input captions in the same file as video.
- If your captions are in a separate file, which you specify in your input captions selector, see Supported captions workflows, sidecar input captions.

Supported captions workflows, input captions in the same file as video

To look up whether MediaConvert supports your captions workflow, choose the topic from the following list that corresponds to your video output container. Then find the row that corresponds to your input captions container and format. Check the supported output captions format column to see the available output captions formats.

Topics

- Supported captions in MXF output container
- Supported captions in QuickTime output container
- Sidecar captions supported with File output groups

Supported captions in CMAF output container

The following table lists supported output captions formats for this output container. Use when your input captions are in the same container or stream as your video. This table is sorted by the input captions container and input captions format.

If your input captions are in a sidecar format, see Sidecar captions supported in CMAF output container. *Sidecar captions* are captions that you provide as a separate input file from your video.

Input captions container	Input captions format	Supported output captions formats
HLS Container	Embedded	IMSC (as sidecar .fmp4) WebVTT
	WebVTT	IMSC (as sidecar .fmp4) WebVTT
MP4 Container	Embedded	IMSC (as sidecar .fmp4) WebVTT
	SCTE-20	IMSC (as sidecar .fmp4) WebVTT
MPEG2-TS Container	Embedded	IMSC (as sidecar .fmp4) WebVTT
	SCTE-20	IMSC (as sidecar .fmp4) WebVTT

Input captions container	Input captions format	Supported output captions formats
	Teletext	IMSC (as sidecar .fmp4) WebVTT
MXF Container	Embedded	IMSC (as sidecar .fmp4) WebVTT
	Ancillary	IMSC (as sidecar .fmp4) WebVTT
QuickTime Container	Embedded	IMSC (as sidecar .fmp4) WebVTT
	Ancillary	IMSC (as sidecar .fmp4) WebVTT

In the preceding table, **Embedded** and **Ancillary** refer to groups of similar captions formats.

Embedded captions include these formats:

- CEA-608
- EIA-608
- CEA-708
- EIA-708

Ancillary captions include these formats:

- Captions in the QuickTime Captions Track
- Captions in the MXF container VANC data

Supported captions in DASH output container

The following table lists supported output captions formats for this output container. Use when your input captions are in the same container or stream as your video. This table is sorted by the input captions container and input captions format.

If your input captions are in a sidecar format, see Sidecar captions supported in DASH output container. *Sidecar captions* are captions that you provide as a separate input file from your video.

Input captions container	Input captions format	Supported output captions formats
HLS Container	Embedded	Burn in IMSC (as sidecar .fmp4) IMSC (as sidecar .xml) TTML (as sidecar .fmp4) TTML (as sidecar .ttml) WebVTT
	WebVTT	Burn in IMSC (as sidecar .fmp4) IMSC (as sidecar .xml) TTML (as sidecar .fmp4) TTML (as sidecar .ttml) WebVTT
IMF Container	IMSC1 text profile	Burn in IMSC (as sidecar .fmp4) IMSC (as sidecar .xml)

Input captions container	Input captions format	Supported output captions formats
Note Specify the CPL to define your input.		TTML (as sidecar .fmp4) TTML (as sidecar .ttml)
MP4 Container	Embedded	Burn in IMSC (as sidecar .fmp4) IMSC (as sidecar .xml) TTML (as sidecar .fmp4) TTML (as sidecar .ttml) WebVTT
	SCTE-20	Burn in IMSC (as sidecar .fmp4) IMSC (as sidecar .xml) TTML (as sidecar .fmp4) TTML (as sidecar .ttml)
MPEG2-TS Container	DVB-Sub	Burn in

Input captions container	Input captions format	Supported output captions formats
	Embedded	Burn in IMSC (as sidecar .fmp4) IMSC (as sidecar .xml) TTML (as sidecar .fmp4) TTML (as sidecar .ttml) WebVTT
	SCTE-20	Burn in IMSC (as sidecar .fmp4) IMSC (as sidecar .xml) TTML (as sidecar .fmp4) TTML (as sidecar .ttml) WebVTT
	Teletext	Burn in IMSC (as sidecar .fmp4) IMSC (as sidecar .xml) TTML (as sidecar .fmp4) TTML (as sidecar .ttml)

Input captions container	Input captions format	Supported output captions formats
MXF Container	Embedded	Burn in IMSC (as sidecar .fmp4) IMSC (as sidecar .xml) TTML (as sidecar .fmp4) TTML (as sidecar .ttml)
	SCTE-20	Burn in IMSC (as sidecar .fmp4) IMSC (as sidecar .xml) TTML (as sidecar .fmp4) TTML (as sidecar .ttml)
	Ancillary	Burn in IMSC (as sidecar .fmp4) IMSC (as sidecar .xml) TTML (as sidecar .fmp4) TTML (as sidecar .ttml)
	Teletext	Burn in IMSC (as sidecar .fmp4) IMSC (as sidecar .xml) TTML (as sidecar .fmp4) TTML (as sidecar .ttml)

Input captions container	Input captions format	Supported output captions formats
QuickTime Container	Embedded	Burn in IMSC (as sidecar .fmp4) IMSC (as sidecar .xml) TTML (as sidecar .fmp4) TTML (as sidecar .ttml)
	SCTE-20	Burn in IMSC (as sidecar .fmp4) IMSC (as sidecar .xml) TTML (as sidecar .fmp4) TTML (as sidecar .ttml)
	Ancillary	Burn in IMSC (as sidecar .fmp4) IMSC (as sidecar .xml) TTML (as sidecar .fmp4) TTML (as sidecar .ttml)

In the preceding table, **Embedded** and **Ancillary** refer to groups of similar captions formats.

Embedded captions include these formats:

- CEA-608
- EIA-608
- CEA-708
- EIA-708

Ancillary captions include these formats:

- Captions in the QuickTime Captions Track
- Captions in the MXF container VANC data

Supported captions in HLS output container

The following table lists supported output captions formats for this output container. Use when your input captions are in the same container or stream as your video. This table is sorted by the input captions container and input captions format.

If your input captions are in a sidecar format, see Sidecar captions supported in HLS output container. *Sidecar captions* are captions that you provide as a separate input file from your video.

> **Note**
>
> AWS Elemental MediaConvert has the following limitations with Teletext in outputs:
>
> - The service doesn't support captions formatting and positioning
> - You can use only Teletext level 1.5 languages

Input captions container	Input captions format	Supported output captions formats
HLS Container	Embedded	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded WebVTT
	WebVTT	Burn in WebVTT
IMF Container	IMSC1 text profile	WebVTT

Input captions container	Input captions format	Supported output captions formats
Note Specify the CPL to define your input.		
MP4 Container	Embedded	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded WebVTT
	SCTE-20	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded WebVTT
MPEG2-TS Container	Embedded	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded WebVTT

Input captions container	Input captions format	Supported output captions formats
	SCTE-20	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded WebVTT
	Teletext	Burn in WebVTT
	DVB-Sub	Burn in
MXF Container	Embedded	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded WebVTT
	SCTE-20	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded WebVTT

Input captions container	Input captions format	Supported output captions formats
	Ancillary	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded WebVTT
	Teletext	Burn in
QuickTime Container	Embedded	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded WebVTT
	SCTE-20	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded WebVTT

Input captions container	Input captions format	Supported output captions formats
	Ancillary	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded WebVTT

In the preceding table, **Embedded** and **Ancillary** refer to groups of similar captions formats.

Embedded captions include these formats:

- CEA-608
- EIA-608
- CEA-708
- EIA-708

Ancillary captions include these formats:

- Captions in the QuickTime Captions Track
- Captions in the MXF container VANC data

Supported captions in Microsoft Smooth Streaming (MSS) output container

The following table lists supported output captions formats for this output container. Use when your input captions are in the same container or stream as your video. This table is sorted by the input captions container and input captions format.

If your input captions are in a sidecar format, see Sidecar captions supported in Microsoft Smooth Streaming (MSS) output container. *Sidecar captions* are captions that you provide as a separate input file from your video.

Input captions container	Input captions format	Supported output captions formats
HLS Container	Embedded	Burn in TTML
	WebVTT	Burn in TTML (as sidecar .ttml)
MP4 Container	Embedded	Burn in TTML
	SCTE-20	Burn in TTML
MPEG2-TS Container	Embedded	Burn in TTML
	SCTE-20	Burn in TTML
	Teletext	Burn in TTML
	DVB-Sub	Burn in
MXF Container	Embedded	Burn in TTML
	SCTE-20	Burn in TTML
	Ancillary	Burn in

Input captions container	Input captions format	Supported output captions formats
		TTML
	Teletext	Burn in TTML
QuickTime Container	Embedded	Burn in TTML
	SCTE-20	Burn in TTML
	Ancillary	Burn in TTML

In the preceding table, **Embedded** and **Ancillary** refer to groups of similar captions formats.

Embedded captions include these formats:

- CEA-608
- EIA-608
- CEA-708
- EIA-708

Ancillary captions include these formats:

- Captions in the QuickTime Captions Track
- Captions in the MXF container VANC data

Supported captions in MP4 output container

The following table lists supported output captions formats for this output container. Use when your input captions are in the same container or stream as your video. This table is sorted by the input captions container and input captions format.

If your input captions are in a sidecar format, see Sidecar captions supported in MP4 output container. *Sidecar captions* are captions that you provide as a separate input file from your video.

Input captions container	Input captions format	Supported output captions formats
HLS Container	Embedded	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded
	WebVTT	Burn in
MP4 Container	Embedded	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded
	SCTE-20	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded
MPEG2-TS Container	Embedded	Burn in Embedded

Input captions container	Input captions format	Supported output captions formats
		Embedded plus SCTE-20 SCTE-20 plus embedded
	SCTE-20	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded
	Teletext	Burn in
	DVB-Sub	Burn in
MXF Container	Embedded	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded
	SCTE-20	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded
	Ancillary	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded

Input captions container	Input captions format	Supported output captions formats
	Teletext	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded
QuickTime Container	Embedded	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded
	SCTE-20	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded
	Ancillary	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded

In the preceding table, **Embedded** and **Ancillary** refer to groups of similar captions formats.

Embedded captions include these formats:

- CEA-608
- EIA-608

- CEA-708
- EIA-708

Ancillary captions include these formats:

- Captions in the QuickTime Captions Track
- Captions in the MXF container VANC data

Supported captions in MPEG2-TS output container

The following table lists supported output captions formats for this output container. Use when your input captions are in the same container or stream as your video. This table is sorted by the input captions container and input captions format.

If your input captions are in a sidecar format, see Sidecar captions supported in MPEG2-TS File output container. *Sidecar captions* are captions that you provide as a separate input file from your video.

> **Note**
>
> AWS Elemental MediaConvert has the following limitations with Teletext in outputs:
>
> - The service doesn't support captions formatting and positioning
> - You can use only Teletext level 1.5 languages

Input captions container	Input captions format	Supported output captions formats
HLS Container	Embedded	Burn in DVB-Sub Embedded Embedded plus SCTE-20

Input captions container	Input captions format	Supported output captions formats
		SCTE-20 plus embedded
	WebVTT	Burn in Teletext DVB-Sub
IMF Container **Note** Specify the CPL to define your input.	IMSC1 text profile	Burn in DVB-Sub Teletext
MPEG2-TS Container	Embedded	Burn in DVB-Sub Embedded Embedded plus SCTE-20 SCTE-20 plus embedded Teletext
	SCTE-20	Burn in DVB-Sub Embedded Embedded plus SCTE-20 SCTE-20 plus embedded

Input captions container	Input captions format	Supported output captions formats
	Teletext	Burn in DVB-Sub Teletext
	DVB-Sub	Burn in DVB-Sub
MP4 Container	Embedded	Burn in DVB-Sub Embedded Embedded plus SCTE-20 SCTE-20 plus embedded Teletext
	SCTE-20	Burn in DVB-Sub Embedded Embedded plus SCTE-20 SCTE-20 plus embedded

Input captions container	Input captions format	Supported output captions formats
MXF Container	Embedded	Burn in DVB-Sub Embedded Embedded plus SCTE-20 SCTE-20 plus embedded Teletext
	SCTE-20	Burn in DVB-Sub Embedded Embedded plus SCTE-20 SCTE-20 plus embedded
	Ancillary	Burn in DVB-Sub Embedded Embedded plus SCTE-20 SCTE-20 plus embedded
	Teletext	Burn in DVB-Sub Teletext

Input captions container	Input captions format	Supported output captions formats
QuickTime Container	Embedded	Burn in DVB-Sub Embedded Embedded plus SCTE-20 SCTE-20 plus embedded
	SCTE-20	Burn in DVB-Sub Embedded Embedded plus SCTE-20 SCTE-20 plus embedded
	Ancillary	Burn in DVB-Sub Embedded Embedded plus SCTE-20 SCTE-20 plus embedded

In the preceding table, **Embedded** and **Ancillary** refer to groups of similar captions formats.

Embedded captions include these formats:

- CEA-608
- EIA-608
- CEA-708
- EIA-708

Ancillary captions include these formats:

- Captions in the QuickTime Captions Track
- Captions in the MXF container VANC data

Supported captions in MXF output container

The following table lists supported output captions formats for this output container. Use when your input captions are in the same container or stream as your video. This table is sorted by the input captions container and input captions format.

If your input captions are in a sidecar format, see Sidecar captions supported in MXF output container. *Sidecar captions* are captions that you provide as a separate input file from your video.

> **Note**
>
> AWS Elemental MediaConvert has the following limitations with Teletext in outputs:
>
> - The service doesn't support captions formatting and positioning
> - You can use only Teletext level 1.5 languages

Input captions container	Input captions format	Supported output captions formats
HLS Container	Embedded	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded
	WebVTT	Burn in Teletext
IMF Container	IMSC1 text profile	Burn in

Input captions container	Input captions format	Supported output captions formats
Note Specify the CPL to define your input.		
MP4 Container	Embedded	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded Teletext
	SCTE-20	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded
MPEG2-TS Container	DVB-Sub	Burn in
	Embedded	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded Teletext

Input captions container	Input captions format	Supported output captions formats
	SCTE-20	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded
	Teletext	Burn in Teletext
MXF Container	Ancillary	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded
	Embedded	Embedded Teletext
QuickTime Container	Embedded	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded
	SCTE-20	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded

Input captions container	Input captions format	Supported output captions formats
	Ancillary	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded

In the preceding table, **Embedded** and **Ancillary** refer to groups of similar captions formats.

Embedded captions include these formats:

- CEA-608
- EIA-608
- CEA-708
- EIA-708

Ancillary captions include these formats:

- Captions in the QuickTime Captions Track
- Captions in the MXF container VANC data

Supported captions in QuickTime output container

The following table lists supported output captions formats for this output container. Use when your input captions are in the same container or stream as your video. This table is sorted by the input captions container and input captions format.

If your input captions are in a sidecar format, see Sidecar captions supported in QuickTime output container. *Sidecar captions* are captions that you provide as a separate input file from your video.

Input captions container	Input captions format	Supported output captions formats
HLS Container	Embedded	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded
	WebVTT	Burn in
IMF Container **Note** Specify the CPL to define your input.	IMSC1 text profile	Burn in
MP4 Container	Embedded	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded
	SCTE-20	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded
MPEG2-TS Container	Embedded	Burn in Embedded

Input captions container	Input captions format	Supported output captions formats
		Embedded plus SCTE-20 SCTE-20 plus embedded
	SCTE-20	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded
	Teletext	Burn in
	DVB-Sub	Burn in
MXF Container	Embedded	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded
	SCTE-20	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded

Input captions container	Input captions format	Supported output captions formats
	Ancillary	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded
	Teletext	Burn in
QuickTime Container	Embedded	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded
	SCTE-20	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded
	Ancillary	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded

In the preceding table, **Embedded** and **Ancillary** refer to groups of similar captions formats.

Embedded captions include these formats:

- CEA-608
- EIA-608
- CEA-708
- EIA-708

Ancillary captions include these formats:

- Captions in the QuickTime Captions Track
- Captions in the MXF container VANC data

Sidecar captions supported with File output groups

The following table lists standalone sidecar output captions formats that MediaConvert supports with outputs in the **File** output group. *Sidecar* captions are captions that are in a separate file from your video.

If your input captions are in a sidecar format, see Sidecar captions supported with File output groups. *Sidecar captions* are captions that you provide as a separate input file from your video.

When you set up these output captions in your job, choose **No container** (`RAW`) for **Container**, under **Output settings**. In your JSON job specification, specify it this way:

```
{
           "ContainerSettings": {
             "Container": "RAW"
           },
```

> **Note**
>
> You can create sidecar captions outputs only as part of a job that also generates a video output.

Input captions container	Input captions format	Supported output captions formats
HLS Container	Embedded	IMSC (as sidecar .xml) SCC SRT SMI TTML WebVTT
MP4 Container	Embedded	IMSC (as sidecar .xml) SCC SRT SMI TTML WebVTT
	SCTE-20	IMSC (as sidecar .xml) SCC SRT SMI TTML WebVTT
MPEG2-TS Container	Embedded	IMSC (as sidecar .xml) SCC

Input captions container	Input captions format	Supported output captions formats
		SRT SMI TTML WebVTT
	SCTE-20	IMSC (as sidecar .xml) SCC SRT SMI TTML WebVTT
	Teletext	IMSC (as sidecar .xml) SCC SRT SMI TTML WebVTT

Input captions container	Input captions format	Supported output captions formats
MXF Container	Embedded	IMSC (as sidecar .xml) SCC SRT SMI TTML WebVTT
	SCTE-20	IMSC (as sidecar .xml) SCC SRT SMI TTML WebVTT
	Ancillary	IMSC (as sidecar .xml) SCC SRT SMI TTML WebVTT

Input captions container	Input captions format	Supported output captions formats
	Teletext	IMSC (as sidecar .xml) SCC SRT SMI TTML WebVTT
QuickTime Container	Embedded	IMSC (as sidecar .xml) SCC SRT SMI TTML WebVTT
	SCTE-20	IMSC (as sidecar .xml) SCC SRT SMI TTML WebVTT

Input captions container	Input captions format	Supported output captions formats
	Ancillary	IMSC (as sidecar .xml) SCC SRT SMI TTML WebVTT

In the preceding table, **Embedded** and **Ancillary** refer to groups of similar captions formats.

Embedded captions include these formats:

- CEA-608
- EIA-608
- CEA-708
- EIA-708

Ancillary captions include these formats:

- Captions in the QuickTime Captions Track
- Captions in the MXF container VANC data

Supported captions workflows, sidecar input captions

The following tables show the captions formats you can create in your outputs when your input captions are in a sidecar format. *Sidecar captions* are captions that you provide as a separate input file from your video.

To see whether MediaConvert supports your captions workflow, go to the table for your video output container. MediaConvert doesn't support every possible file extension for each sidecar format.

Topics

Sidecar captions supported in CMAF output container

The following table lists supported output captions formats for this output container when your input captions are in a sidecar format. *Sidecar* captions are captions that are in a separate file from your video.

Input captions format	Supported file extensions	Supported output captions formats
IMSC1 text profile	.xml	IMSC (as sidecar .fmp4) WebVTT
SCC	.scc	IMSC (as sidecar .fmp4) WebVTT
SMI	.smi	IMSC (as sidecar .fmp4) WebVTT
SMPTE-TT	.ttml, .xml, .dfxp	IMSC (as sidecar .fmp4) WebVTT
SRT	.srt	IMSC (as sidecar .fmp4)

Input captions format	Supported file extensions	Supported output captions formats
		WebVTT
EBU STL	.stl	IMSC (as sidecar .fmp4) WebVTT
TTML	.ttml, .xml, .dfxp	IMSC (as sidecar .fmp4) WebVTT
WebVTT	.vtt	IMSC (as sidecar .fmp4) WebVTT

Sidecar captions supported in DASH output container

The following table lists supported output captions formats for this output container when your input captions are in a sidecar format. *Sidecar* captions are captions that are in a separate file from your video.

Input captions format	Supported file extensions	Supported output captions formats
IMSC1 text profile	.xml	Burn in IMSC (as sidecar .fmp4) IMSC (as sidecar .xml) TTML (as sidecar .fmp4) TTML (as sidecar .ttml)
SCC	.scc	Burn in IMSC (as sidecar .fmp4) IMSC (as sidecar .xml)

Input captions format	Supported file extensions	Supported output captions formats
		TTML (as sidecar .fmp4) TTML (as sidecar .ttml)
SMI	.smi	Burn in IMSC (as sidecar .fmp4) IMSC (as sidecar .xml) TTML (as sidecar .fmp4) TTML (as sidecar .ttml)
SMPTE-TT	.ttml, .xml, .dfxp	Burn in IMSC (as sidecar .fmp4) IMSC (as sidecar .xml) TTML (as sidecar .fmp4) TTML (as sidecar .ttml)
SRT	.srt	Burn in IMSC (as sidecar .fmp4) IMSC (as sidecar .xml) TTML (as sidecar .fmp4) TTML (as sidecar .ttml)

Input captions format	Supported file extensions	Supported output captions formats
EBU STL	.stl	Burn in IMSC (as sidecar .fmp4) IMSC (as sidecar .xml) TTML (as sidecar .fmp4) TTML (as sidecar .ttml)
TTML	.ttml, .xml, .dfxp	Burn in IMSC (as sidecar .fmp4) IMSC (as sidecar .xml) TTML (as sidecar .fmp4) TTML (as sidecar .ttml)
WebVTT	.vtt	Burn in IMSC (as sidecar .fmp4) IMSC (as sidecar .xml) TTML (as sidecar .fmp4) TTML (as sidecar .ttml)

Sidecar captions supported in HLS output container

The following table lists supported output captions formats for this output container when your input captions are in a sidecar format. *Sidecar* captions are captions that are in a separate file from your video.

Input captions format	Supported file extensions	Supported output captions formats
IMSC1 text profile	.xml	Burn in WebVTT
SCC	.scc	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded WebVTT
SMI	.smi	Burn in WebVTT
SMPTE-TT	.ttml, .xml, .dfxp	Burn in WebVTT
SRT	.srt	Burn in WebVTT
EBU STL	.stl	Burn in WebVTT
TTML	.ttml, .xml, .dfxp	Burn in WebVTT
WebVTT	.vtt	Burn in WebVTT

Sidecar captions supported in Microsoft Smooth Streaming (MSS) output container

The following table lists supported output captions formats for this output container when your input captions are in a sidecar format. *Sidecar* captions are captions that are in a separate file from your video.

Input captions format	Supported file extensions	Supported output captions formats
IMSC1 text profile	.xml	Burn in TTML
SCC	.scc	Burn in TTML
SMI	.smi	Burn in TTML
SMPTE-TT	.ttml, .xml, .dfxp	Burn in TTML
SRT	.srt	Burn in TTML
EBU STL	.stl	Burn in TTML
TTML	.ttml, .xml, .dfxp	Burn in TTML
WebVTT	.vtt	Burn in TTML

Sidecar captions supported in MP4 output container

The following table lists supported output captions formats for this output container when your input captions are in a sidecar format. *Sidecar* captions are captions that are in a separate file from your video.

Input captions format	Supported file extensions	Supported output captions formats
IMSC1 text profile	.xml	Burn in
SCC	.scc	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded
SMI	.smi	Burn in
SMPTE-TT	.ttml, .xml, .dfxp	Burn in TTML
SRT	.srt	Burn in
EBU STL	.stl	Burn in
TTML	.ttml, .xml, .dfxp	Burn in
WebVTT	.vtt	Burn in

Sidecar captions supported in MPEG2-TS File output container

The following table lists supported output captions formats for this output container when your input captions are in a sidecar format. *Sidecar* captions are captions that are in a separate file from your video.

Input captions format	Supported file extensions	Supported output captions formats
IMSC1 text profile	.xml	Burn in Teletext DVB-Sub
SCC	.scc	Burn in DVB-Sub Embedded Embedded plus SCTE-20 SCTE-20 plus embedded Teletext
SMI	.smi	Burn in DVB-Sub
SMPTE-TT	.ttml, .xml, .dfxp	Burn in Teletext DVB-Sub
SRT	.srt	Burn in Teletext
EBU STL	.stl	Burn in Teletext DVB-Sub
TTML	.ttml, .xml, .dfxp	Burn in

Input captions format	Supported file extensions	Supported output captions formats
		Teletext DVB-Sub
WebVTT	.vtt	Burn in Teletext DVB-Sub

Sidecar captions supported in MXF output container

The following table lists supported output captions formats for this output container when your input captions are in a sidecar format. *Sidecar* captions are captions that are in a separate file from your video.

Input captions format	Supported file extensions	Supported output captions formats
IMSC1 text profile	.xml	Burn in Teletext
SCC	.scc	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded Teletext
SMI	.smi	Burn in
SMPTE-TT	.ttml, .xml, .dfxp	Burn in
SRT	.srt	Burn in

Input captions format	Supported file extensions	Supported output captions formats
		Teletext
EBU STL	.stl	Burn in Teletext
TTML	.ttml, .xml, .dfxp	Burn in Teletext
WebVTT	.vtt	Burn in Teletext

Sidecar captions supported in QuickTime output container

The following table lists supported output captions formats for this output container when your input captions are in a sidecar format. *Sidecar* captions are captions that are in a separate file from your video.

Input captions format	Supported file extensions	Supported output captions formats
IMSC1 text profile	.xml	Burn in
SCC	.scc	Burn in Embedded Embedded plus SCTE-20 SCTE-20 plus embedded
SMI	.smi	Burn in
SMPTE-TT	.ttml, .xml, .dfxp	Burn in

Input captions format	Supported file extensions	Supported output captions formats
SRT	.srt	Burn in
EBU STL	.stl	Burn in
TTML	.ttml, .xml, .dfxp	Burn in
WebVTT	.vtt	Burn in

Sidecar captions supported with File output groups

The following table lists standalone sidecar output captions formats that MediaConvert supports with outputs in the **File** output group. *Sidecar* captions are captions that are in a separate file from your video.

When you set up these output captions in your job, choose **No container** (RAW) for **Container**, under **Output settings**. In your JSON job specification, specify it this way:

```
{
          "ContainerSettings": {
            "Container": "RAW"
          },
```

> **Note**
>
> You can create sidecar captions outputs only as part of a job that also generates a video output.

Input captions format	Supported file extensions	Supported output captions formats
IMSC1 text profile	.xml	IMSC (as sidecar .xml) IMSC SRT

Input captions format	Supported file extensions	Supported output captions formats
		SMI TTML WebVTT
SCC	.scc	IMSC (as sidecar .xml) SCC SRT SMI TTML WebVTT
SMI	.smi	IMSC (as sidecar .xml) SRT SMI TTML WebVTT
SMPTE-TT	.ttml, .xml, .dfxp	IMSC (as sidecar .xml) SRT SMI TTML WebVTT

Input captions format	Supported file extensions	Supported output captions formats
SRT	.srt	IMSC (as sidecar .xml) IMSC SRT SMI TTML WebVTT
EBU STL	.stl	IMSC (as sidecar .xml) SRT SMI TTML WebVTT
TTML	.ttml, .xml, .dfxp	IMSC (as sidecar .xml) SRT SMI TTML WebVTT

Input captions format	Supported file extensions	Supported output captions formats
WebVTT	.vtt	IMSC (as sidecar .xml) SRT SMI TTML WebVTT

IMSC captions support in AWS Elemental MediaConvert

Current support for IMSC captions workflows is restricted as follows:

- Text profile is the only supported IMSC profile.
- All output IMSC files are IMSC 1.1.

Working with jobs

A job does the work of transcoding a media file. When you create a job, you specify the information that AWS Elemental MediaConvert requires to perform the transcoding. For example, you specify which files to transcode, what to name the transcoded files and where to save them, which advanced features to apply, and so on.

Topics

Creating a job

To create a job, you specify your input settings, output settings, and any job-wide settings. For a detailed step-by-step procedure, see Configuring jobs in MediaConvert. The following procedure is a high level overview of how to create a job using the AWS Management Console.

When you create a job, you submit it to a queue for processing. Processing begins automatically from your queues as resources allow. For information about resource allocation, see Multiple queues and performance testing.

To create a job using the MediaConvert console

1. Open the Jobs page in the MediaConvert console.
2. Choose **Create job**.

3. On the **Create job** page, provide transcode instructions and job settings. For more information, see Configuring jobs in MediaConvert.

   Make sure that you select the same Region for your job and your file storage.

4. Choose **Create**.

You can also create a job using a Template, Preset, duplicated job, or job settings JSON.

Duplicating a job

To create a job that is similar to one that you ran before, you can duplicate a job from your job history. You can also modify any settings if you want to change them.

To create a job based on a recent job using the MediaConvert console

1. Open the Jobs page in the MediaConvert console.
2. Choose the **Job ID** of the job that you want to duplicate.
3. Choose **Duplicate**.
4. Optionally modify any job settings.

   Settings that are likely to change from job to job include the following: input file location, output destination locations, and output name modifiers. If you run transcoding jobs for your customers who have different AWS accounts from your account, you also must change the **IAM role** under **Job settings**.

5. Choose **Create** at the bottom of the page.

Exporting and importing jobs

Completed MediaConvert jobs remain on the **Jobs** page for three months. If you want to be able to run a new job based on a completed job more than three months after you run it, export the job after it is complete and save it. Depending on how many jobs that you run, exporting and then importing a job can be simpler than finding a particular job in your list and duplicating it.

To export a job using the MediaConvert console

1. Open the Jobs page in the MediaConvert console.

2. Choose the **Job ID** of the job that you want to export.
3. On the **Job summary** page, choose the **View JSON** button.
4. Choose **Copy** to copy the JSON to your clipboard.
5. Paste into your JSON editor and save.

To import a job using the MediaConvert console

1. Open the Jobs page in the MediaConvert console.
2. Choose **Import job**.

Viewing your job history

You can view the recent history of MediaConvert jobs that you created with your AWS account in a given AWS Region. After three months, the service automatically deletes the record of a job.

The **Jobs** page shows jobs that are successfully completed, are canceled, are being processed, are waiting in the queue, and that ended in error. You can filter the job history list by the status and by the queue that the jobs were sent to. You can also choose a specific job from the list to view the job's settings.

Console

To view your jobs using the MediaConvert console

1. Open the Jobs page in the MediaConvert console.
2. Optionally, filter the list by status and queue by choosing from the dropdown lists.
3. To see details for a job, choose a **Job ID** to view its **Job summary** page.

CLI

The following `list-jobs` example lists up to twenty of your most recently created jobs.

```
aws mediaconvert list-jobs
```

For more information about how to cancel a job using the AWS CLI, see the AWS CLI command reference.

Searching for jobs

You can search recent jobs that you created with your AWS account in a given AWS Region according to input file name. After three months, the service automatically deletes the record of a job.

Console

To search your jobs by using the MediaConvert console:

1. Open the Search page in the MediaConvert console.
2. Optionally, filter the list by queue, job status, or input file name.
3. To see details for a job, choose a **Job ID** to view its **Job summary** page.

CLI

The following `search-jobs` example searches for a job with an input file name *my-video.mp4* in the `Default` queue with a status of `COMPLETE`. Note that `status` can be `SUBMITTED`, `PROGRESSING`, `COMPLETE`, `CANCELED`, or `ERROR`. Leave `status` or `queue` out to search all job statuses or queues:

```
aws mediaconvert search-jobs \
 --input-file my-video.mp4 \
 --status COMPLETE \
 --queue Default
```

Optionally, you can also search for any part of the input file name string:

```
aws mediaconvert search-jobs --input-file s3://DOC-EXAMPLE-BUCKET/my-video.mp4
aws mediaconvert search-jobs --input-file my-video
```

The AWS CLI will return a JSON similar to the following example:

```
{
    "jobs": [
        {
            "arn": "arn:aws:mediaconvert:us-west-2:111122223333:jobs/1671543211799-
abchg1",
```

```
            "id": "1671543211799-abchg1",
            "createdAt": 1715666761,
            "queue": "arn:aws:mediaconvert:us-west-2:111122223333:queues/Default",
            "status": "COMPLETE",
            "timing": {
                "submitTime": 1715666861,
                "startTime": 1715666863,
                "finishTime": 1715666900
            },
            "accelerationSettings": {
                "mode": "DISABLED"
            },
            "accelerationStatus": "NOT_APPLICABLE",
            "messages": {
                "info": [],
                "warning": []
            },
            "warnings": []
        }
    ]
}
```


For more information about how to search jobs by using the AWS CLI, see the AWS CLI command reference.

Canceling a job

The following procedure explains how to cancel a job using the AWS Elemental MediaConvert console.

Console

To cancel a job using the MediaConvert console

1. Open the Jobs page in the MediaConvert console.
2. Select the **Job ID** of the job that you want to cancel by choosing the option (○) next to it.
3. Choose **Cancel job**.

CLI

The following `cancel-job` example cancels a job.

```
aws mediaconvert cancel-job \
 --id 1234567890123-efg456
```

For more information about how to cancel a job using the AWS CLI, see the AWS CLI command reference.

Configuring jobs in MediaConvert

A job does the work of transcoding a media file into packages and files in various formats and in different sizes for distribution to end viewers. When you create a job, you specify the information that the service requires to perform the transcoding. This includes which file to transcode, which types of files to create, where to store files, which advanced features to apply, and more.

To configure a job, you define input files for the service to transcode, and you specify the source for each video, audio, and captions media element. That source might be a specific part of the primary input file, or it might be a separate file. Next, you specify the types of output files and packages that you want AWS Elemental MediaConvert to generate from the input. You also specify the detailed encoding settings to produce the quality and type of output that you want.

This tutorial shows how to configure jobs in MediaConvert to transcode media files into different formats.

Topics

Optional step: Pause queues

If you're a new customer or you're experimenting with the MediaConvert console, you can pause your queues to avoid accidentally starting a job before you're ready. For more information about queues, see Queues.

To pause or reactivate an on-demand queue using the AWS Management Console

1. Open the Queues page in the MediaConvert console.
2. On the **Queues** page, choose the name of the queue that you want to pause or reactivate.
3. On the queue’s page, choose the **Edit queue** button.
4. On the **Edit queue** page, for **Status**, choose **Paused** or **Active**.
5. Choose **Save queue**.

Step 1: Specify input files

The first part of setting up a MediaConvert job is specifying the location of your input file or files.

To specify the location of your input

1. Open the MediaConvert console at https://console.aws.amazon.com/mediaconvert.
2. On the **Create job** page, in the **Job** pane on the left, choose **Input 1**.
3. In the **Input 1** pane, provide the URI to your video input file that is stored in Amazon S3 or on an HTTP(S) server. For Amazon S3 inputs, you can specify the URI directly or choose **Browse** to select from your Amazon S3 buckets. For HTTP(S) inputs, provide the URL to your input video file. For more information, see HTTP input requirements.

   > **Note**
   >
   > If your input audio or captions are in a separate file, don't create separate inputs for them. You specify these files later in this procedure, within your audio and captions selectors.

4. To join more than one input file into a single asset (input stitching), add another input to the job. To do so, in the **Job** pane, in the **Inputs** section, choose **Add**.

For jobs that have multiple input files, MediaConvert creates outputs by concatenating the inputs in the order that you specify them in the job. You can include up to 150 inputs in your job.

> **Tip**
>
> You can transcode portions of your inputs. For more information, see Input settings.

Step 2: Create input selectors for video, audio, and captions

Next, create input selectors to flag the video, audio, and captions elements from your input that you will use in your outputs. This labels each input element so that you can point to it when you set up your outputs. When you set up input selectors, you also provide the service with information about where to find the data and how to interpret it.

To set up your input selectors

1. In the **Video selector** section, specify values for the fields that are applicable to your job.

   You don't need to create a video selector because MediaConvert automatically creates a video selector when you begin setting up a job. However, the service doesn't automatically detect information about the video source. You can provide this information in the **Video selector** fields. If you keep these settings in their default state, you will create a valid job. For more information about individual settings, choose the **Info** link next to each setting.

   > **Note**
   >
   > MediaConvert doesn't support inputs with multiple video streams, such as Quad 4k. Each input can have only one video selector. Therefore, there is no **Add video selector** button on the console.

2. In the **Audio selectors** section, under **Audio selector 1**, specify information about your primary audio asset. You don't need to create an audio selector 1 because the service automatically creates the first audio selector when you set up a job.

> **Note**
>
> An *audio asset* is often dialogue, background sound, and music together in one track. Tracks often consist of multiple channels. For example, Dolby 5.1 sound has six channels per track.

a. For **Selector type**, choose the way that your audio assets are identified. Often, this is by track. If you are using an HLS input, and would like to select an alternate audio rendition, see Using alternate audio renditions.

b. Provide the identifier (that is, track number, PID, or language code) for your primary audio asset. Your primary audio asset is likely to be track 1.

> **Note**
>
> For most use cases, you associate one input track per input selector. If your use case requires combining multiple tracks into one track, or multiple tracks into one rendition of a streaming package, combine multiple input tracks in one audio selector by typing a comma-separated list. For more information about combining tracks, see Setting up audio tracks and audio selectors.

c. If your audio is in a separate file from your video, choose the **External file** slider switch element and provide the URI to your audio input file that is stored in Amazon S3 or on an HTTP(S) server. For Amazon S3 inputs, you can specify the URI directly or choose **Browse** to select from your Amazon S3 buckets. For HTTP(S) inputs, provide the URL to your input video file. For more information, see HTTP input requirements.

3. If you have additional audio assets, such as multiple language tracks, choose **Add audio selector**. Then provide information about the next asset that is described in the preceding step of this procedure.

4. In the **Captions selectors** section, choose **Add captions selector**. This creates input captions selectors for any sets of captions that you plan to use in an output. For more information about setting up captions for your job, see Setting up captions in AWS Elemental MediaConvert jobs.

Step 3: Create output groups

After specifying your input, you create output groups. The choices that you make when you set up output groups affect the types of assets that your job produces and which devices can play them.

You can use MediaConvert to create media assets that fall broadly into two categories:

- **ABR streaming packages**. You can create adaptive bitrate (ABR) packages so that end viewers can download the asset gradually while they watch. Depending on how you set up your outputs, the end viewer's device can adapt to changes in the available bandwidth by downloading higher or lower-quality segments. ABR packages are also called ABR *stacks*, because they are made up of a stack of video, audio, and captions components. Each component in the stack or package is called a *rendition*.
- **Standalone files**. You might create these files and host them in a location where end viewers download the entire file all at once and then view it. You might also create standalone files and then send them to downstream systems for packaging and distribution.

To create an output group

1. In the **Job** pane, in the **Output groups** section, choose **Add**.
2. Choose an output group type, and then choose **Select**.

   Create one file output group for all the standalone files that you intend to create. Create one ABR streaming output group for each ABR streaming package that you intend to create. For guidance on which ABR streaming output groups to include in your job, see Choosing your ABR streaming output groups.
3. Optionally, for **Custom group name**, enter a name for your group. Any name that you provide here appears in the **Output groups** section of the console but does not affect your outputs.
4. For **Destination**, specify the URI for the Amazon S3 location where the transcoding service will store your output files. You can specify the URI directly or choose **Browse** to select from your Amazon S3 buckets.

   > **Note**
   >
   > You can optionally append a basename to your destination URI. To create the file name of your final asset, the transcoding service uses this basename and any name modifier that you provide in the individual output settings.

If you don't provide a basename with your URI, the transcoding service generates a basename from the input 1 file name, minus the extension.

5. Specify the values for any additional settings that apply to the entire output group. These settings vary depending on the type of output group that you select. For more information about individual settings, choose the **Info** link next to each setting.

Step 4: Create outputs

After you create output groups, set up your outputs in each group. The number of outputs for each output group depends on the output group type, as follows:

- For **File** output groups, include all elements of the media asset in one output. This includes any audio or captions that you provide in a separate file.
- For ABR streaming output groups—**CMAF**, **Apple HLS**, **DASH ISO**, and **Microsoft Smooth Streaming**—create a separate output for each media element. That is, one output per video resolution, one output per audio track, and one output per captions language.

Choose from one of the following procedures that correspond to the output group types that you created in Step 3: Create output groups.

Creating outputs in ABR streaming output groups

For each ABR streaming output group that you set up in Step 3: Create output groups, create and set up an output for each media element that you want in the ABR streaming package.

Creating video ABR streaming outputs

For each video output that you include in your output group, MediaConvert creates one video rendition, or set of segmented video files. Multiple video renditions in a streaming package, of varying resolutions and video quality, allow the end viewer's device to adapt the quality of video to the available bandwidth.

Note

Although the job has only one video *input* selector, ABR streaming output groups often have several video *outputs* per output group.

To create and set up video ABR streaming outputs

1. On the **Create job** page, in the **Job** pane on the left, under **Output Groups**, below the **CMAF**, **Apple HLS**, **DASH ISO**, or **Microsoft Smooth Streaming** output group that you want add outputs to, choose **Output 1**.

   When you create an output group, MediaConvert automatically populates the output group with output 1. You don't need to explicitly create the first output.

2. In the **Output settings** pane, for **Name modifier**, enter a value.

   MediaConvert appends the name modifier to the file names that it creates for this output. Enter a name modifier that will make it easy to identify which files came from which output, such as `-video-hi-res`.

3. If one of the predefined groups of settings listed under **Preset** is suitable for your workflow, choose it from the list. If you use a preset, skip the next step of this procedure.

4. Specify your video settings as follows:

   a. In the **Output settings** section, specify values for any remaining general settings. Depending on the output group type, these settings might include transport stream settings or other container settings. For more information about individual settings, choose the **Info** link next to each setting.

   b. In the **Stream settings** section, specify values for video encoding. The video settings are selected by default, so you don't need to explicitly choose this group of settings.

      There is only one input video selector per job, so you don't need to explicitly choose it when you set up your video outputs.

   For more information about individual settings, choose the **Info** links on the console.

5. If your output includes a group of audio settings by default, delete it as follows:

   a. In the **Stream settings** section, choose **Audio 1**.

   b. Choose **Remove audio**.

6. If you want multiple video renditions in your ABR streaming package, repeat the preceding steps of this procedure. This will create an additional video output for each one.

Creating audio ABR streaming outputs

For each audio output that you include in your output group, MediaConvert creates one audio rendition, or set of segmented video files. The most common reason to include multiple audio renditions is to provide multiple language options. If you provide only one language, you probably need only one audio output.

> **Note**
>
> For AAC streaming outputs, the initial segment is longer in duration than the others. This is because, with AAC, the initial segment must contain silent AAC pre-roll samples before the audible part of the segment. MediaConvert accounts for these extra samples in the timestamps, so the audio plays back correctly.

To create and set up audio ABR streaming outputs

1. If you're working in a CMAF output group, skip this step. The first audio output is created for you.

   Create an output for your first audio track. Usually an audio track corresponds to one language.

   a. In the **Job** pane, choose the output group that you're working in.

   b. In the **Outputs** pane, choose **Add output**.

   c. Choose the output that you just created.

   d. If your output includes a group of video settings by default, choose **Remove video** to delete it. This k the **Audio 1** group of settings displayed.

2. In the **Output settings** pane, for **Name modifier**, enter a value.

   MediaConvert appends the name modifier to the file names that it creates for this output. Enter a name modifier that will make it easy to identify which files came from which output, such as `-audio-english`.

3. If one of the predefined groups of settings listed under **Preset** is suitable for your workflow, choose it from the list. If you use a preset, skip the next step of this procedure.

4. Specify your audio settings as follows:

   a. In the **Output settings** section, specify values for any remaining general settings. For more information about individual settings, choose the **Info** link next to each setting.
   b. Under **Stream settings**, for **Audio source**, choose one of the audio selectors that you created in Step 2: Create input selectors for video, audio, and captions.
   c. In the **Stream settings** section, specify values for audio encoding. For more information about individual settings, choose the **Info** link next to each setting.

5. If you have additional audio assets to include in the ABR streaming package, create an output for each of them as follows:

   a. In the **Job** pane, choose the output group that you're working in.
   b. In the **Outputs** pane, choose **Add output**.
   c. Choose the output that you just created.
   d. If your output includes a group of video settings by default, choose **Remove video** to delete it. This keeps the **Audio 1** group of settings displayed.
   e. Set up the output as described in steps 2 through 4 of this procedure.

Creating captions for ABR streaming outputs

Setting up captions can be complex. For detailed information, see Setting up captions in AWS Elemental MediaConvert jobs. For basic instructions, complete the following procedure.

To create and set up captions for ABR streaming outputs

1. Create an output for your first set of captions. Usually a set of captions corresponds to one language.

   a. In the **Job** pane, choose the output group that you're working in.
   b. In the **Outputs** pane, choose **Add output**.
   c. Choose the output that you just created.
   d. If your output includes groups of video and audio settings by default, choose **Remove video** and **Remove audio** to delete them.
   e. Choose **Add captions** to display a set of captions settings.

2. In the **Output settings** pane, for **Name modifier**, enter a value.

MediaConvert appends the name modifier to the file names that it creates for this output. Enter a name modifier that will make it easy to identify which files came from which output, such as `-captions-english`.

3. Specify your captions settings as follows:

   a. In the **Output settings** section, specify values for any remaining general settings. For more information about individual settings, choose the **Info** link next to each setting.

   b. Under **Stream settings**, for **Captions source**, choose one of the captions selectors that you created in Step 2: Create input selectors for video, audio, and captions.

   c. In the **Stream settings** section, specify values for the remaining captions settings.

Creating additional manifests

By default, MediaConvert generates a single top-level manifest for each of your CMAF, DASH ISO, Apple HLS, and Microsoft Smooth Streamingoutput groups. This default manifest references all the outputs in the output group.

Optionally, you can create additional top-level manifests that reference only a subset of the outputs in your output group. For example, you might want to create a manifest that doesn't include HDR outputs, for viewers who don't have a subscription that includes HDR.

> **Note**
>
> For CMAF output groups, if you keep the default enabled value for **Write HLS manifest** and **Write DASH manifest**, MediaConvert creates additional manifests in both of those formats. If you disable either of those settings, MediaConvert doesn't create additional manifests in that format.

To create an additional manifest

1. On the **Create job** page, in the **Job** pane on the left, choose the output group that you want to create the additional manifest for.
2. In the **Additional manifests** section on the right, choose **Add manifest**.
3. For **Manifest name modifier**, enter the text that you want to be at the end of the manifest file name, before the extension. This setting is required, because it gives each manifest a different file name.

4. For **Select outputs**, choose the outputs that you want the manifest to refer to.
5. Repeat these steps to create up to 10 additional manifests. Each additional manifest must have a different value for **Manifest name modifier**.

Creating and setting up outputs in File output groups

With File output groups, each asset that the service creates corresponds to one output, rather than one output group. Each asset contains all video, audio, and captions elements. Therefore, it's simplest to set up by first creating the output, and then setting up all the output selectors.

Create file outputs

If you created a file output group in Step 3: Create output groups, create and set up an output in the file output group for each standalone file that you intend to create.

To create an output in a file output group

1. When you create an output group, MediaConvert automatically populates the output group with output 1, so you don't need to explicitly create it. If you are creating only one standalone file, skip the rest of this procedure.
2. If you want to create more than one standalone file, create additional outputs as follows:

    a. On the **Create job** page, in the **Job** pane on the left, under **Output Groups**, choose **File group**.

    b. In the **Outputs** pane, choose **Add output**.

Set up output selectors in file outputs

Next, for each file output that you just created, set up output selectors.

To set up output selectors in a file output

1. On the **Create job** page, in the **Job** pane on the left, under **Output Groups**, under **File group**, choose **Output 1**.
2. In the **Output settings** pane, for **Name modifier**, enter a value.

    MediaConvert appends the name modifier to the file names that it creates for this output. Enter a name modifier that identifies which files came from which output, such as `-standalone-hi-res`.

3. If one of the predefined groups of settings listed under **Preset** is suitable for your workflow, choose it from the list. If you use a preset, skip step 4 of this procedure.

   Output presets can contain up to one set each of video, audio, and captions settings. Therefore, if your standalone output file contains more than one audio or captions asset, you can't use a preset. If you can't use presets in your output, but you want to use the preset settings as a starting point, choose the preset, then choose **No preset** from the **Preset** dropdown list. This prepopulates your output with the same settings that are in the preset.

4. Specify your output settings as follows:

   a. In the **Output settings** section, specify values for any remaining general settings. These settings vary depending on the container that you choose. For more information about individual settings, choose the **Info** link next to each setting.

   b. In the **Stream settings** section, specify values for video encoding. For more information about individual settings, choose the **Info** link next to each setting.

      Note

      The video settings tab is selected by default, so you don't need to explicitly choose this group of settings. There is only one input video selector per job, so you don't need to explicitly choose it when you set up your video outputs.

   c. Choose **Audio 1** to display the group of encoding settings for the first audio asset. **Audio 1** is located on the left side of the **Stream settings** pane, below **Video**.

   d. Under **Stream settings**, for **Audio source**, choose one of the audio selectors that you created in Step 2: Create input selectors for video, audio, and captions.

   e. In the **Stream settings** section, specify values for audio encoding. For more information about individual settings, choose the **Info** link next to each setting.

   f. To include captions in the output, choose **Add captions**. This displays a group of captions settings. For more information about setting up captions, see Setting up captions in AWS Elemental MediaConvert jobs.

Step 5: Specify global job settings

Global job settings apply to every output that the job creates.

If your job incorporates audio or captions provided in a separate file from your input, or if you use the graphic overlay (image inserter) feature, it is especially important to get these settings right.

There are three distinct groups of timecode settings. Global job timecode configuration is one of those three. For more information about the different sets of timecode settings and how MediaConvert manages timecodes, see Setting up timecodes.

To specify global job settings

1. In the **Job** pane, in the **Job settings** section, choose **AWS integration**.
2. For **IAM role**, choose an IAM role that has permissions to access the Amazon S3 buckets that hold your input and output files. The IAM role must have a trusted relationship with MediaConvert. For information about creating this role, see Setting up IAM permissions .
3. Optionally, specify job-wide timecode settings in the **Timecode configuration** pane.
4. Specify values for the other job settings and enable global processors. For more information about individual settings, choose the **Info** link next to each setting.

Example AWS Elemental MediaConvert job settings in JSON

The job settings in these examples represent the simplest valid jobs you can run. They work well for experimenting with the service. When you want to perform more complex transcodes or create different outputs, use the console to set up your job and to generate your JSON job specification. To do so, in the **Job** pane on the left, under **Job settings**, choose **Show job JSON**.

For more information about submitting your job programmatically, see one of the introductory topics of the *AWS Elemental MediaConvert API Reference*:

- Getting started with AWS Elemental MediaConvert using the AWS SDKs or the AWS CLI
- Getting started with AWS Elemental MediaConvert using the API

Important

We recommend that you use the MediaConvert console to generate your production JSON job specification.
Your job specification must conform to validation by the transcoding engine. The transcoding engine validations represent complex dependencies among groups of settings and dependencies between your transcoding settings and properties of your input files.

The MediaConvert console functions as an interactive job builder to make it easy to create valid job JSON specifications. You can use job templates and output presets to get started quickly.

To use these examples, replace the following placeholder values with actual values:

- ROLE HERE
- s3://DOC-EXAMPLE-BUCKET
- s3://DOC-EXAMPLE-BUCKET1

Examples

- Example—mp4 output
- Example—ABR output
- Example—automated ABR

Example—mp4 output

```
{
  "UserMetadata": {},
  "Role": "ROLE ARN",
  "Settings": {
    "OutputGroups": [
      {
        "Name": "File Group",
        "OutputGroupSettings": {
          "Type": "FILE_GROUP_SETTINGS",
          "FileGroupSettings": {
            "Destination": "s3://&example-s3-bucket1;/out"
          }
        },
        "Outputs": [
          {
            "VideoDescription": {
              "ScalingBehavior": "DEFAULT",
              "TimecodeInsertion": "DISABLED",
              "AntiAlias": "ENABLED",
              "Sharpness": 50,
```

```
                "CodecSettings": {
                  "Codec": "H_264",
                  "H264Settings": {
                    "InterlaceMode": "PROGRESSIVE",
                    "NumberReferenceFrames": 3,
                    "Syntax": "DEFAULT",
                    "Softness": 0,
                    "GopClosedCadence": 1,
                    "GopSize": 48,
                    "Slices": 1,
                    "GopBReference": "DISABLED",
                    "SlowPal": "DISABLED",
                    "SpatialAdaptiveQuantization": "ENABLED",
                    "TemporalAdaptiveQuantization": "ENABLED",
                    "FlickerAdaptiveQuantization": "DISABLED",
                    "EntropyEncoding": "CABAC",
                    "Bitrate": 4500000,
                    "FramerateControl": "SPECIFIED",
                    "RateControlMode": "CBR",
                    "CodecProfile": "HIGH",
                    "Telecine": "NONE",
                    "MinIInterval": 0,
                    "AdaptiveQuantization": "HIGH",
                    "CodecLevel": "LEVEL_4_1",
                    "FieldEncoding": "PAFF",
                    "SceneChangeDetect": "ENABLED",
                    "QualityTuningLevel": "SINGLE_PASS_HQ",
                    "FramerateConversionAlgorithm": "DUPLICATE_DROP",
                    "UnregisteredSeiTimecode": "DISABLED",
                    "GopSizeUnits": "FRAMES",
                    "ParControl": "INITIALIZE_FROM_SOURCE",
                    "NumberBFramesBetweenReferenceFrames": 3,
                    "RepeatPps": "DISABLED",
                    "HrdBufferSize": 9000000,
                    "HrdBufferInitialFillPercentage": 90,
                    "FramerateNumerator": 24000,
                    "FramerateDenominator": 1001
                  }
                },
                "AfdSignaling": "NONE",
                "DropFrameTimecode": "ENABLED",
                "RespondToAfd": "NONE",
                "ColorMetadata": "INSERT",
                "Width": 1920,
```

```
            "Height": 1080
          },
          "AudioDescriptions": [
            {
              "AudioTypeControl": "FOLLOW_INPUT",
              "CodecSettings": {
                "Codec": "AAC",
                "AacSettings": {
                  "AudioDescriptionBroadcasterMix": "NORMAL",
                  "Bitrate": 96000,
                  "RateControlMode": "CBR",
                  "CodecProfile": "LC",
                  "CodingMode": "CODING_MODE_2_0",
                  "RawFormat": "NONE",
                  "SampleRate": 48000,
                  "Specification": "MPEG4"
                }
              },
              "LanguageCodeControl": "FOLLOW_INPUT"
            }
          ],
          "ContainerSettings": {
            "Container": "MP4",
            "Mp4Settings": {
              "CslgAtom": "INCLUDE",
              "FreeSpaceBox": "EXCLUDE",
              "MoovPlacement": "PROGRESSIVE_DOWNLOAD"
            }
          }
        }
      ]
    }
  ],
  "AdAvailOffset": 0,
  "Inputs": [
    {
      "AudioSelectors": {
        "Audio Selector 1": {
          "Tracks": [
            1
          ],
          "Offset": 0,
          "DefaultSelection": "DEFAULT",
          "SelectorType": "TRACK",
```

```
            "ProgramSelection": 1
          },
          "Audio Selector 2": {
            "Tracks": [
              2
            ],
            "Offset": 0,
            "DefaultSelection": "NOT_DEFAULT",
            "SelectorType": "TRACK",
            "ProgramSelection": 1
          }
        },
        "VideoSelector": {
          "ColorSpace": "FOLLOW"
        },
        "FilterEnable": "AUTO",
        "PsiControl": "USE_PSI",
        "FilterStrength": 0,
        "DeblockFilter": "DISABLED",
        "DenoiseFilter": "DISABLED",
        "TimecodeSource": "EMBEDDED",
        "FileInput": "s3://&example-s3-bucket;"
      }
    ]
  }
}
```

Example—ABR output

```
{
  "UserMetadata": {},
  "Role": "ROLE ARN",
  "Settings": {
    "OutputGroups": [
      {
        "Name": "Apple HLS",
        "Outputs": [
          {
            "ContainerSettings": {
              "Container": "M3U8",
              "M3u8Settings": {
                "AudioFramesPerPes": 2,
```

```
                "PcrControl": "PCR_EVERY_PES_PACKET",
                "PmtPid": 480,
                "PrivateMetadataPid": 503,
                "ProgramNumber": 1,
                "PatInterval": 100,
                "PmtInterval": 100,
                "VideoPid": 481,
                "AudioPids": [
                  482,
                  483,
                  484,
                  485,
                  486,
                  487,
                  488,
                  489,
                  490,
                  491,
                  492
                ]
              }
            },
            "VideoDescription": {
              "Width": 1920,
              "Height": 1080,
              "VideoPreprocessors": {
                "Deinterlacer": {
                  "Algorithm": "INTERPOLATE",
                  "Mode": "DEINTERLACE"
                }
              },
              "AntiAlias": "ENABLED",
              "Sharpness": 100,
              "CodecSettings": {
                "Codec": "H_264",
                "H264Settings": {
                  "InterlaceMode": "PROGRESSIVE",
                  "ParNumerator": 1,
                  "NumberReferenceFrames": 3,
                  "Softness": 0,
                  "FramerateDenominator": 1001,
                  "GopClosedCadence": 1,
                  "GopSize": 90,
                  "Slices": 1,
```

```
                "HrdBufferSize": 12500000,
                "ParDenominator": 1,
                "SpatialAdaptiveQuantization": "ENABLED",
                "TemporalAdaptiveQuantization": "DISABLED",
                "FlickerAdaptiveQuantization": "DISABLED",
                "EntropyEncoding": "CABAC",
                "Bitrate": 8500000,
                "FramerateControl": "SPECIFIED",
                "RateControlMode": "CBR",
                "CodecProfile": "HIGH",
                "Telecine": "NONE",
                "FramerateNumerator": 30000,
                "MinIInterval": 0,
                "AdaptiveQuantization": "MEDIUM",
                "CodecLevel": "LEVEL_4",
                "SceneChangeDetect": "ENABLED",
                "QualityTuningLevel": "SINGLE_PASS_HQ",
                "GopSizeUnits": "FRAMES",
                "ParControl": "SPECIFIED",
                "NumberBFramesBetweenReferenceFrames": 3,
                "HrdBufferInitialFillPercentage": 90,
                "Syntax": "DEFAULT"
              }
            },
            "AfdSignaling": "NONE",
            "DropFrameTimecode": "ENABLED",
            "RespondToAfd": "NONE",
            "ColorMetadata": "INSERT"
          },
          "AudioDescriptions": [
            {
              "AudioTypeControl": "FOLLOW_INPUT",
              "AudioSourceName": "Audio Selector 1",
              "CodecSettings": {
                "Codec": "AAC",
                "AacSettings": {
                  "Bitrate": 128000,
                  "RateControlMode": "CBR",
                  "CodecProfile": "LC",
                  "CodingMode": "CODING_MODE_2_0",
                  "SampleRate": 48000
                }
              },
              "LanguageCodeControl": "FOLLOW_INPUT"
```

```
          }
        ],
        "NameModifier": "_high"
      },
      {
        "VideoDescription": {
          "ScalingBehavior": "DEFAULT",
          "TimecodeInsertion": "DISABLED",
          "AntiAlias": "ENABLED",
          "Sharpness": 50,
          "CodecSettings": {
            "Codec": "H_264",
            "H264Settings": {
              "InterlaceMode": "PROGRESSIVE",
              "NumberReferenceFrames": 3,
              "Syntax": "DEFAULT",
              "Softness": 0,
              "GopClosedCadence": 1,
              "GopSize": 90,
              "Slices": 1,
              "GopBReference": "DISABLED",
              "SlowPal": "DISABLED",
              "SpatialAdaptiveQuantization": "ENABLED",
              "TemporalAdaptiveQuantization": "ENABLED",
              "FlickerAdaptiveQuantization": "DISABLED",
              "EntropyEncoding": "CABAC",
              "Bitrate": 7500000,
              "FramerateControl": "INITIALIZE_FROM_SOURCE",
              "RateControlMode": "CBR",
              "CodecProfile": "MAIN",
              "Telecine": "NONE",
              "MinIInterval": 0,
              "AdaptiveQuantization": "HIGH",
              "CodecLevel": "AUTO",
              "FieldEncoding": "PAFF",
              "SceneChangeDetect": "ENABLED",
              "QualityTuningLevel": "SINGLE_PASS",
              "FramerateConversionAlgorithm": "DUPLICATE_DROP",
              "UnregisteredSeiTimecode": "DISABLED",
              "GopSizeUnits": "FRAMES",
              "ParControl": "INITIALIZE_FROM_SOURCE",
              "NumberBFramesBetweenReferenceFrames": 2,
              "RepeatPps": "DISABLED"
            }
```

```
            },
            "AfdSignaling": "NONE",
            "DropFrameTimecode": "ENABLED",
            "RespondToAfd": "NONE",
            "ColorMetadata": "INSERT",
            "Width": 1280,
            "Height": 720
          },
          "AudioDescriptions": [
            {
              "AudioTypeControl": "FOLLOW_INPUT",
              "CodecSettings": {
                "Codec": "AAC",
                "AacSettings": {
                  "AudioDescriptionBroadcasterMix": "NORMAL",
                  "Bitrate": 96000,
                  "RateControlMode": "CBR",
                  "CodecProfile": "LC",
                  "CodingMode": "CODING_MODE_2_0",
                  "RawFormat": "NONE",
                  "SampleRate": 48000,
                  "Specification": "MPEG4"
                }
              },
              "LanguageCodeControl": "FOLLOW_INPUT"
            }
          ],
          "OutputSettings": {
            "HlsSettings": {
              "AudioGroupId": "program_audio",
              "AudioRenditionSets": "program_audio",
              "IFrameOnlyManifest": "EXCLUDE"
            }
          },
          "ContainerSettings": {
            "Container": "M3U8",
            "M3u8Settings": {
              "AudioFramesPerPes": 4,
              "PcrControl": "PCR_EVERY_PES_PACKET",
              "PmtPid": 480,
              "PrivateMetadataPid": 503,
              "ProgramNumber": 1,
              "PatInterval": 0,
              "PmtInterval": 0,
```

```
                "Scte35Source": "NONE",
                "Scte35Pid": 500,
                "TimedMetadata": "NONE",
                "TimedMetadataPid": 502,
                "VideoPid": 481,
                "AudioPids": [
                  482,
                  483,
                  484,
                  485,
                  486,
                  487,
                  488,
                  489,
                  490,
                  491,
                  492
                ]
              }
            },
            "NameModifier": "_med"
          },
          {
            "VideoDescription": {
              "ScalingBehavior": "DEFAULT",
              "TimecodeInsertion": "DISABLED",
              "AntiAlias": "ENABLED",
              "Sharpness": 100,
              "CodecSettings": {
                "Codec": "H_264",
                "H264Settings": {
                  "InterlaceMode": "PROGRESSIVE",
                  "NumberReferenceFrames": 3,
                  "Syntax": "DEFAULT",
                  "Softness": 0,
                  "GopClosedCadence": 1,
                  "GopSize": 90,
                  "Slices": 1,
                  "GopBReference": "DISABLED",
                  "SlowPal": "DISABLED",
                  "SpatialAdaptiveQuantization": "ENABLED",
                  "TemporalAdaptiveQuantization": "ENABLED",
                  "FlickerAdaptiveQuantization": "DISABLED",
                  "EntropyEncoding": "CABAC",
```

```
                "Bitrate": 3500000,
                "FramerateControl": "INITIALIZE_FROM_SOURCE",
                "RateControlMode": "CBR",
                "CodecProfile": "MAIN",
                "Telecine": "NONE",
                "MinIInterval": 0,
                "AdaptiveQuantization": "HIGH",
                "CodecLevel": "LEVEL_3_1",
                "FieldEncoding": "PAFF",
                "SceneChangeDetect": "ENABLED",
                "QualityTuningLevel": "SINGLE_PASS_HQ",
                "FramerateConversionAlgorithm": "DUPLICATE_DROP",
                "UnregisteredSeiTimecode": "DISABLED",
                "GopSizeUnits": "FRAMES",
                "ParControl": "INITIALIZE_FROM_SOURCE",
                "NumberBFramesBetweenReferenceFrames": 2,
                "RepeatPps": "DISABLED"
              }
            },
            "AfdSignaling": "NONE",
            "DropFrameTimecode": "ENABLED",
            "RespondToAfd": "NONE",
            "ColorMetadata": "INSERT",
            "Width": 960,
            "Height": 540
          },
          "AudioDescriptions": [
            {
              "AudioTypeControl": "FOLLOW_INPUT",
              "CodecSettings": {
                "Codec": "AAC",
                "AacSettings": {
                  "AudioDescriptionBroadcasterMix": "NORMAL",
                  "Bitrate": 96000,
                  "RateControlMode": "CBR",
                  "CodecProfile": "LC",
                  "CodingMode": "CODING_MODE_2_0",
                  "RawFormat": "NONE",
                  "SampleRate": 48000,
                  "Specification": "MPEG4"
                }
              },
              "LanguageCodeControl": "FOLLOW_INPUT"
            }
```

```
          ],
          "OutputSettings": {
            "HlsSettings": {
              "AudioGroupId": "program_audio",
              "AudioRenditionSets": "program_audio",
              "IFrameOnlyManifest": "EXCLUDE"
            }
          },
          "ContainerSettings": {
            "Container": "M3U8",
            "M3u8Settings": {
              "AudioFramesPerPes": 4,
              "PcrControl": "PCR_EVERY_PES_PACKET",
              "PmtPid": 480,
              "PrivateMetadataPid": 503,
              "ProgramNumber": 1,
              "PatInterval": 0,
              "PmtInterval": 0,
              "Scte35Source": "NONE",
              "Scte35Pid": 500,
              "TimedMetadata": "NONE",
              "TimedMetadataPid": 502,
              "VideoPid": 481,
              "AudioPids": [
                482,
                483,
                484,
                485,
                486,
                487,
                488,
                489,
                490,
                491,
                492
              ]
            }
          },
          "NameModifier": "_low"
        }
      ],
      "OutputGroupSettings": {
        "Type": "HLS_GROUP_SETTINGS",
        "HlsGroupSettings": {
```

```
            "ManifestDurationFormat": "INTEGER",
            "SegmentLength": 10,
            "TimedMetadataId3Period": 10,
            "CaptionLanguageSetting": "OMIT",
            "Destination": "s3://bucket/hls1/master",
            "TimedMetadataId3Frame": "PRIV",
            "CodecSpecification": "RFC_4281",
            "OutputSelection": "MANIFESTS_AND_SEGMENTS",
            "ProgramDateTimePeriod": 600,
            "MinSegmentLength": 0,
            "DirectoryStructure": "SINGLE_DIRECTORY",
            "ProgramDateTime": "EXCLUDE",
            "SegmentControl": "SEGMENTED_FILES",
            "ManifestCompression": "NONE",
            "ClientCache": "ENABLED",
            "StreamInfResolution": "INCLUDE"
          }
        }
      }
    ],
    "AdAvailOffset": 0,
    "Inputs": [
      {
        "AudioSelectors": {
          "Audio Selector 1": {
            "Tracks": [
              1
            ],
            "Offset": 0,
            "DefaultSelection": "DEFAULT",
            "SelectorType": "TRACK",
            "ProgramSelection": 1
          },
          "Audio Selector 2": {
            "Tracks": [
              2
            ],
            "Offset": 0,
            "DefaultSelection": "NOT_DEFAULT",
            "SelectorType": "TRACK",
            "ProgramSelection": 1
          }
        },
        "VideoSelector": {
```

```
          "ColorSpace": "FOLLOW"
        },
        "FilterEnable": "AUTO",
        "PsiControl": "USE_PSI",
        "FilterStrength": 0,
        "DeblockFilter": "DISABLED",
        "DenoiseFilter": "DISABLED",
        "TimecodeSource": "EMBEDDED",
        "FileInput": "s3://INPUT"
      }
    ]
  }
}
```

Example—automated ABR

This example JSON job specification specifies an automated ABR stack in Apple HLS. In addition to specifying the automated ABR settings, it explicitly sets these values:

- Accelerated transcoding `Mode` to `PREFERRED`
- `rateControlMode` to `QVBR`
- `qualityTuningLevel` to `MULTI_PASS_HQ`

For information about the automated ABR settings, see Automated ABR.

```
{
  "UserMetadata": {},
  "Role": "ROLE ARN",
  "Settings": {
    "TimecodeConfig": {
      "Source": "ZEROBASED"
    },
    "OutputGroups": [
      {
        "Name": "Apple HLS",
        "Outputs": [
          {
            "ContainerSettings": {
              "Container": "M3U8",
              "M3u8Settings": {
                "AudioFramesPerPes": 4,
```

```
            "PcrControl": "PCR_EVERY_PES_PACKET",
            "PmtPid": 480,
            "PrivateMetadataPid": 503,
            "ProgramNumber": 1,
            "PatInterval": 0,
            "PmtInterval": 0,
            "Scte35Source": "NONE",
            "NielsenId3": "NONE",
            "TimedMetadata": "NONE",
            "VideoPid": 481,
            "AudioPids": [
              482,
              483,
              484,
              485,
              486,
              487,
              488,
              489,
              490,
              491,
              492
            ]
          }
        },
        "VideoDescription": {
          "ScalingBehavior": "DEFAULT",
          "TimecodeInsertion": "DISABLED",
          "AntiAlias": "ENABLED",
          "Sharpness": 50,
          "CodecSettings": {
            "Codec": "H_264",
            "H264Settings": {
              "InterlaceMode": "PROGRESSIVE",
              "NumberReferenceFrames": 3,
              "Syntax": "DEFAULT",
              "Softness": 0,
              "FramerateDenominator": 1,
              "GopClosedCadence": 1,
              "GopSize": 60,
              "Slices": 2,
              "GopBReference": "DISABLED",
              "EntropyEncoding": "CABAC",
              "FramerateControl": "SPECIFIED",
```

```
                "RateControlMode": "QVBR",
                "CodecProfile": "MAIN",
                "Telecine": "NONE",
                "FramerateNumerator": 30,
                "MinIInterval": 0,
                "AdaptiveQuantization": "AUTO",
                "CodecLevel": "AUTO",
                "FieldEncoding": "PAFF",
                "SceneChangeDetect": "ENABLED",
                "QualityTuningLevel": "MULTI_PASS_HQ",
                "FramerateConversionAlgorithm": "DUPLICATE_DROP",
                "UnregisteredSeiTimecode": "DISABLED",
                "GopSizeUnits": "FRAMES",
                "ParControl": "INITIALIZE_FROM_SOURCE",
                "NumberBFramesBetweenReferenceFrames": 2,
                "RepeatPps": "DISABLED",
                "DynamicSubGop": "STATIC"
              }
            },
            "AfdSignaling": "NONE",
            "DropFrameTimecode": "ENABLED",
            "RespondToAfd": "NONE",
            "ColorMetadata": "INSERT"
          },
          "OutputSettings": {
            "HlsSettings": {
              "AudioGroupId": "program_audio",
              "AudioRenditionSets": "program_audio",
              "AudioOnlyContainer": "AUTOMATIC",
              "IFrameOnlyManifest": "EXCLUDE"
            }
          },
          "NameModifier": "video"
        },
        {
          "ContainerSettings": {
            "Container": "M3U8",
            "M3u8Settings": {
              "AudioFramesPerPes": 4,
              "PcrControl": "PCR_EVERY_PES_PACKET",
              "PmtPid": 480,
              "PrivateMetadataPid": 503,
              "ProgramNumber": 1,
              "PatInterval": 0,
```

```
            "PmtInterval": 0,
            "Scte35Source": "NONE",
            "NielsenId3": "NONE",
            "TimedMetadata": "NONE",
            "TimedMetadataPid": 502,
            "VideoPid": 481,
            "AudioPids": [
              482,
              483,
              484,
              485,
              486,
              487,
              488,
              489,
              490,
              491,
              492
            ]
          }
        },
        "AudioDescriptions": [
          {
            "AudioTypeControl": "FOLLOW_INPUT",
            "AudioSourceName": "Audio Selector 1",
            "CodecSettings": {
              "Codec": "AAC",
              "AacSettings": {
                "AudioDescriptionBroadcasterMix": "NORMAL",
                "Bitrate": 96000,
                "RateControlMode": "CBR",
                "CodecProfile": "LC",
                "CodingMode": "CODING_MODE_2_0",
                "RawFormat": "NONE",
                "SampleRate": 48000,
                "Specification": "MPEG4"
              }
            },
            "LanguageCodeControl": "FOLLOW_INPUT"
          }
        ],
        "OutputSettings": {
          "HlsSettings": {
            "AudioGroupId": "program_audio",
```

```
                "AudioTrackType": "ALTERNATE_AUDIO_AUTO_SELECT_DEFAULT",
                "AudioOnlyContainer": "AUTOMATIC",
                "IFrameOnlyManifest": "EXCLUDE"
              }
            },
            "NameModifier": "audio"
          }
        ],
        "OutputGroupSettings": {
          "Type": "HLS_GROUP_SETTINGS",
          "HlsGroupSettings": {
            "ManifestDurationFormat": "FLOATING_POINT",
            "SegmentLength": 10,
            "TimedMetadataId3Period": 10,
            "CaptionLanguageSetting": "OMIT",
            "Destination": "s3://&example-s3-bucket1;/main",
            "TimedMetadataId3Frame": "PRIV",
            "CodecSpecification": "RFC_4281",
            "OutputSelection": "MANIFESTS_AND_SEGMENTS",
            "ProgramDateTimePeriod": 600,
            "MinSegmentLength": 0,
            "MinFinalSegmentLength": 0,
            "DirectoryStructure": "SINGLE_DIRECTORY",
            "ProgramDateTime": "EXCLUDE",
            "SegmentControl": "SEGMENTED_FILES",
            "ManifestCompression": "NONE",
            "ClientCache": "ENABLED",
            "AudioOnlyHeader": "INCLUDE",
            "StreamInfResolution": "INCLUDE"
          }
        },
        "AutomatedEncodingSettings": {
          "AbrSettings": {
            "MaxRenditions": 6,
            "MaxAbrBitrate": 5000000,
            "MinAbrBitrate": 300000
          }
        }
      }
    ],
    "AdAvailOffset": 0,
    "Inputs": [
      {
        "AudioSelectors": {
```

```
          "Audio Selector 1": {
            "Offset": 0,
            "DefaultSelection": "DEFAULT",
            "ProgramSelection": 1
          }
        },
        "VideoSelector": {
          "ColorSpace": "FOLLOW",
          "Rotate": "DEGREE_0",
          "AlphaBehavior": "DISCARD"
        },
        "FilterEnable": "AUTO",
        "PsiControl": "USE_PSI",
        "FilterStrength": 0,
        "DeblockFilter": "DISABLED",
        "DenoiseFilter": "DISABLED",
        "InputScanType": "AUTO",
        "TimecodeSource": "ZEROBASED",
        "FileInput": "s3://&example-s3-bucket;/test.mov"
      }
    ]
  },
  "AccelerationSettings": {
    "Mode": "PREFERRED"
  },
  "StatusUpdateInterval": "SECONDS_60",
  "Priority": 0
}
```

Specifying input files and input clips

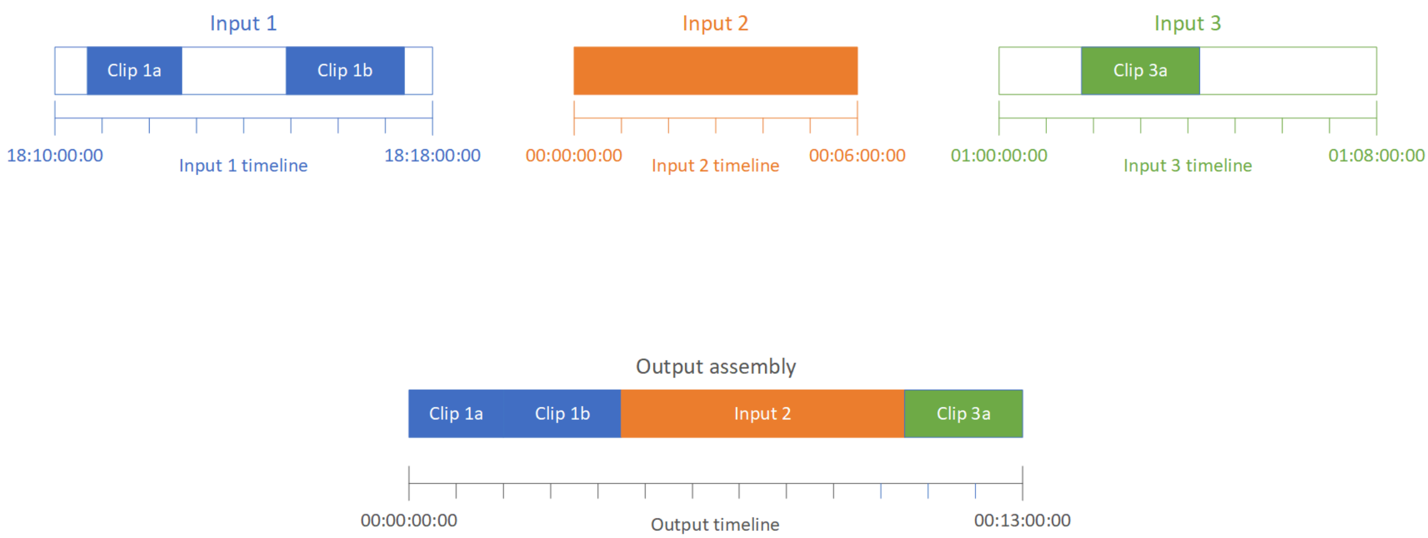
You can use MediaConvert for *assembly workflows*. An assembly workflow is a MediaConvert job that performs basic input clipping and stitching to assemble output assets from different sources without requiring separate editing software. For example, an assembly workflow can put together a bumper followed by feature content that is interleaved with advertisements. The feature content might have a logo graphic overlay at the beginning of each feature segment.

With these kinds of jobs, you assemble your outputs from multiple inputs by using *input stitching*, or portions of inputs by using *input clipping*. MediaConvert creates all of a job's outputs from this assembly. If you want outputs with different clips of the input files or with different arrangements of the inputs, you must create a separate job for each assembly.

Topics

How MediaConvert uses timelines to assemble jobs

MediaConvert assembles inputs and input clips according to *input timelines* and the *output timeline*. The service constructs these timelines based on your settings, and then assembles your inputs into outputs based on them. The following illustration shows three independent input timelines and an output timeline.



Input timelines

Each input has its own *input timeline*. An input timeline is a series of timecodes that MediaConvert generates to represent each frame of the input file.

By default, the input timeline is the same as any timecodes embedded in the input video. You can specify a different starting timecode in the input setting **Timecode source**. If you use the API or an SDK, you can find this setting in the JSON file of your job. The setting name is `TimecodeSource`, located in `Settings`, `Inputs`. For more information, see the section called "Adjusting the input timeline with the input timecode source".

MediaConvert uses the input timeline for the following:

- Determining when input graphic overlays (inserted images) appear in the video. For more information about the difference between input and output overlays, see Choosing between input overlay and output overlay.
- Determining when motion graphic overlays (inserted images) appear in the video. For more information about the different types of graphic overlays, see Image insertion.
- Synchronizing your video with *sidecar captions* that are in a timecode-based format. Sidecar captions are captions that you provide as input files that are separate from the video.
- Interpreting the timecodes that you provide when you specify input clips.

Output timeline

The *output timeline* is the series of timecodes that MediaConvert generates to embed in the outputs. MediaConvert also uses the timecodes of the output timeline for features that apply to every output in the job.

By default, the output timeline is the same as any timecodes embedded in the video of your first input file. You can specify a different starting timecode in the job-wide **Timecode configuration** settings under **Job settings**. If you use the API or an SDK, you can find these settings in the JSON file of your job. These settings are under `Settings`, `TimecodeConfig`. For more information, see Adjusting the output timeline with the job-wide timecode configuration.

MediaConvert uses the output timeline for the following:

- Determining which timecodes to embed in the output video, when you enable **Timecode insertion** in your output timecode settings.
- Determining when output overlays (inserted images) appear in the video. For more information about the different types of graphic overlays, see Image insertion.
- Determining how your HLS variant playlists show time.
- Interpreting the timecode that you provide when you specify a value for **Anchor timecode**.

Setting up an assembly workflow job

Follow these steps to set up a job that combines assembly workflow features such as input clipping, input stitching, graphic overlay, and sidecar captions sync. Doing these tasks in this order can make setup easier. In particular, we recommend that you specify your input clips last. This is because each input timeline counts frames from the entire input, not from each individual clip.

This procedure relies on the concept of input and output timelines. For more information, see How MediaConvert uses timelines to assemble jobs.

To set up an assembly workflow job (console)

1. **Specify your video input files.**

   You can have up to 150 inputs in a job. MediaConvert stitches together the inputs in the order that you add them. To use multiple clips from the same input file in chronological order without other inputs in between them, specify the input file only once.

   For full instructions, see the section called “Step 1: Specify input files”.

2. **Set up your audio selectors.**

   In each input, you create audio selectors to map your input audio to your outputs. For instructions, see Step 2: Create input selectors for video, audio, and captions.

   With sidecar audio files, MediaConvert synchronizes audio and video without regard to timecodes. MediaConvert lines up the start of the audio file with the start of the video file.

   Whether your audio is in a sidecar file or embedded in the video, you can adjust its sync using the **Offset** setting in the input audio selector. Use a positive number for **Offset** to move the audio later in the input timeline; use a negative number to move it earlier.

3. **Synchronize any sidecar captions.**

   How you set up your sidecar captions sync depends on the input captions format:

   - If your input captions format is timecode-based (for example, SCC or STL), the service synchronizes the timecode in the captions file with the input timeline.
   - If your input captions format is timestamp-based (for example, SRT, SMI, or TTML), the service synchronizes the captions with the video without regard to timecodes.

   Related information

   - About input timecode source and captions alignment
   - the section called “Adjusting the input timeline with the input timecode source”
   - Captions and captions selectors

4. **Set up when you want any graphic overlays or motion graphic overlays to appear.**

How you specify the time that the overlay appears depends on what kind of overlay you specify:

- For input still graphic overlays, specify the overlay in the input where you want the overlay to appear. Specify the start and end times with timecodes that match with that input's timeline.
- For output still graphic overlays, specify when you want the overlay to appear based on the output timeline.
- For motion graphic overlays, specify when you want the overlay to appear based on the inputs' timelines.

Related information

- the section called “Adjusting the input timeline with the input timecode source”
- the section called “Adjusting the output timeline with the job-wide timecode configuration”
- Image insertion

5. **Specify input clips.**

   Unless you want MediaConvert to include the full duration of the input, specify input clips for each input. Specify the start and end times with timecodes that match with that input's timeline.

   Set up input clips as follows:

   a. On the **Create job** page, in the **Job** pane on the left, choose an input.

   b. In the **Input clips** section, choose **Add input clip**.

   c. Enter the starting and ending timecodes for the first clip that you want to include. Use the following 24-hour format with a frame number: HH:MM:SS:FF.

      When you specify an input clip for an audio-only input, the last numbers in the timecode that you enter correspond to hundredths of a second. For example, 00:00:30:75 is the same as 30.75 seconds.

      Make sure that you provide timecodes that align with your input timeline. By default, MediaConvert bases input clipping on timecodes that are embedded in your input video.

How you align your timecodes depends on whether your input video has embedded timecodes:

- If your input doesn't have embedded timecodes, set **Timecode source** to **Start at 0** or **Specified start**.
- If your input *does* have embedded timecodes and you want MediaConvert to use them, for **Timecode source**, keep the default value, **Embedded**. Specify your clip start and end times accordingly.

  For example, if an input **Timecode source** is set to **Embedded** with video embedded timecodes that start at 01:00:00:00, define the start timecode for a clip 30 seconds in 01:00:30:00 (not 00:00:30:00). By default, the input timeline is the same as the timecodes that are embedded in the video. You can change what determines the input timeline by adjusting the input **Timecode source** setting.
- Specify an input clip duration that is less than 12 hours long.

For more information, see Adjusting the input timeline with the input timecode source.

d. Specify any additional clips. Multiple clips must be in chronological order and can't overlap; each **Start timecode** must come after the previous clip's **End timecode**.

   If you specify more than one input clip, they all appear in the output, one after the other, in the order that you specify them.

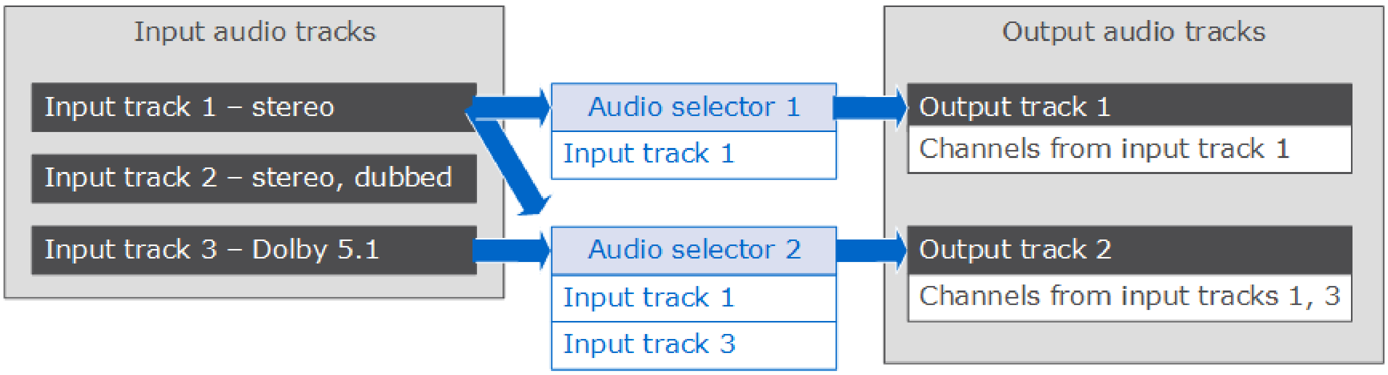
Setting up audio tracks and audio selectors

You use audio selectors to associate input audio with output audio. You can set up a single audio selector to represent one or more tracks from the input. After that, you create audio tracks in the output and associate a single audio selector with each output track.

Associations between input audio tracks, audio selectors, and output audio tracks follow these rules:

- Each input track can be associated with one or more audio selectors
- Each audio selector has one or more input tracks
- Each output track has one audio selector

The following illustration shows these relationships. In the illustration, the input file contains three audio tracks. Audio selector 1 selects input track 1. Audio selector 1 is associated with output audio track 1, so track 1 of the output has the same content as track 1 of the input. The second input audio track is not selected by an audio selector, so it isn't used in the output. Audio selector 2 selects input tracks 1 and 3. Audio selector 2 is associated with output audio track 2, so output track 2 contains the channels from input tracks 1 and 3.



For workflows that require channel-level control, use the audio channel remix feature, which supports the following workflows:

- Changing the order of channels in an audio track
- Moving audio channels from one or more input tracks to different output tracks
- Combining the audio from multiple channels into a single channel
- Splitting the audio from a single channel into multiple channels
- Adjusting the loudness level of audio channels

Setting up captions in AWS Elemental MediaConvert jobs

To include captions in your job, follow these steps in the order listed:

1. If your input captions are a timecode-based sidecar captions format, such as SCC or STL, set the timecode source settings.
2. Gather required captions information.
3. Create input captions selectors.
4. Set up captions in outputs.

For a full list of supported input and output captions, see Captions supported by AWS Elemental MediaConvert.

For information about how to set up captions in your output, see Setting up captions in outputs.

Tip

You can use Amazon Transcribe with MediaConvert to generate captions and include them in your output. For more information, see AWS VOD captioning using Amazon Transcribe in *AWS Samples* on GitHub.

Specifying the timecode source

For your captions to correctly synchronize with your video, you must set up your input timeline to match the timecodes embedded in your captions file. MediaConvert establishes the input timeline based on the value you choose for the input **Timecode source** setting. For more information, see Input timecode source and captions alignment.

For instructions on adjusting the **Timecode source** setting, see Adjusting the input timeline with the input timecode source.

Gathering required captions information

Before you set up captions in your job, note the following information:

- The *input captions format*. You must have this information ahead of time; MediaConvert does not read this from your input files.
- The *tracks* from the input captions that you intend to use in any of your outputs.
- The *output packages and files* that you intend to create with the job. For information about specifying the output package or file type, see Creating outputs.
- The *output captions format* that you intend to use in each output.

  For supported output captions based on your input container, input captions format, and output container, see Supported captions workflows, input captions in the same file as video.
- The *output captions tracks* that you intend to include for each output. If you pass through teletext-to-teletext, all tracks in the input are available in the output. Otherwise, the tracks that you include in an output might be a subset of the tracks that are available in the input.

Creating input captions selectors

When you set up captions, you begin by creating captions selectors. Captions selectors identify a particular captions asset on the input and associate a label with it. The captions asset is either a single track or the set of all tracks contained in the input file, depending on your input captions format. For example, you might add **Captions selector 1** and associate the French captions with it. When you set up an output to include captions, you do so by specifying captions selectors.

To create input captions selectors

1. On the **Create job** page, in the **Job** pane on the left, choose an input.

> **Note**
>
> In jobs with multiple inputs, each input must have the same number of captions selectors. For inputs that don't have captions, create empty captions selectors. For these selectors, for **Source**, choose **Null source**. Remove all captions selectors if no inputs have captions.

2. In the **Captions selectors** section, near the bottom of the page, choose **Add captions selector**.
3. Under **Source**, choose the input captions format.
4. For most formats, more fields appear. Specify the values for these fields as described in the topic that relates to your input captions format. Choose the appropriate topic from the list that follows this procedure.
5. Create more captions selectors as necessary. The number of captions selectors that you need depends on your input captions format. Choose the appropriate topic from the list that follows this procedure.

QuickTime captions track or captions in MXF VANC data (ancillary) input captions

If your input captions are in either of the following formats, the service handles them as "ancillary" data:

- QuickTime captions track (format QTCC)
- MXF VANC data

MediaConvert does not create output captions in these formats, but you can convert them to a supported output format.

For ancillary captions

- Create one captions selector per track that you will use in your outputs.
- In each captions selector, for **Source**, choose **Ancillary**.
- In each captions selector, for **CC channel**, choose the channel number for the track that is associated with the selector.

  For example, the input captions have English in CC channel 1 and Spanish in CC channel 2. To use these captions, create Captions selector 1, and then choose 1 in the **CC channel** dropdown list. Next, create Captions selector 2, and then choose 2 in the **CC channel** dropdown list.

Embedded (CEA/EIA-608, CEA/EIA-708), embedded+SCTE-20, and SCTE-20+embedded input captions

If your input captions are in any of the following formats, the service handles them as "embedded":

- CEA-608
- EIA-608
- CEA-708
- EIA-708

If your input captions have both embedded captions and SCTE-20 captions, and you want both types in your outputs, set up separate input captions selectors for the SCTE-20 and the embedded captions tracks. Set up the SCTE-20 captions selectors the same way that you set up the embedded selectors.

> **Note**
>
> For MXF inputs, your captions are most likely on the ancillary track. Some third-party media analysis tools incorrectly report these captions as 608/708 embedded. For information on setting up ancillary captions, see QuickTime captions track or captions in MXF VANC data (ancillary) input captions.

Number of captions selectors for embedded captions

- If all of your output captions are also an embedded format, create only one captions selector, even if you want to include multiple tracks in the output. With this setup, MediaConvert automatically extracts all tracks and includes them in the output.
- If all of your outputs are in a format that is not embedded, create one captions selector for each track that you want to include in the output.
- If some of your outputs have captions in an embedded format and some of your outputs have captions in a different format, create one captions selector for the outputs with embedded captions. Also create individual selectors for the outputs with other captions that aren't embedded, one for each track that you want in your outputs.

Captions selector fields for embedded captions

Source: Choose **Embedded**

CC channel number: This field specifies the track to extract. Complete as follows:

- If you are doing embedded-to-embedded captions (that is, you create only one captions selector for the input embedded captions), MediaConvert ignores this field, so keep the default value for **CC channel number**.
- If you are converting embedded captions to another format, (that is, you create several captions selectors, one for each track), specify the captions channel number from the input that holds the track that you want. To do that, select the channel number from the dropdown list. For example, select **1** to choose CC1.

> **Note**
>
> MediaConvert doesn't automatically detect which language is in each channel. You can specify that when you set up the output captions, so that MediaConvert passes the language code metadata for the captions channel into the output for downstream use.

DVB-Sub input captions

MediaConvert supports DVB-Sub only in TS inputs.

In most cases, create one captions selector per track. In each selector, specify which track you want by providing the PID or language code.

> **Note**
>
> Don't specify the captions in both the **PID** field and the **Language** dropdown list. Specify one or the other.

If you are doing DVB-sub-to-DVB-sub and you want to pass through all the captions tracks from the input to the output, create one captions selector for all tracks. In this case, keep the **PID** field blank and don't choose any language from the **Language** dropdown list.

Teletext input captions

How you set up your Teletext input captions selectors depends on how you plan to use the captions in your output. You can use Teletext captions in one of the following ways:

- Teletext to Teletext passthrough

  With Teletext passthrough, MediaConvert passes through your input captions unchanged from the input to the output. Captions styling, Teletext page numbers, and non-caption Teletext data appear in your outputs exactly the same as in the input.

  Teletext passthrough is the only way to include Teletext data that isn't captions in your output.
- Teletext to Teletext, page remapping

  If you want the Teletext page numbers on your output to differ from the page numbers on the input, you can remap the content. When you do this, your output captions have plain styling and you lose any Teletext data that isn't captions.
- Teletext to other captions formats

  You can use Teletext input captions to generate output captions in some other formats. To look up which captions you can generate from Teletext inputs, see Captions supported by AWS Elemental MediaConvert.

For information on setting up captions for each of these workflows, see the following topics.

Teletext to Teletext passthrough

When you're doing Teletext to Teletext passthrough, create one input captions selector for the whole set of input captions. Don't specify a value for **Page number**.

For information about setting up the output of this captions workflow, see Teletext to Teletext passthrough.

Teletext to Teletext, page remapping

When the captions format for both your input and output captions is Teletext, and you want your output Teletext page numbers to be different from the input page numbers, create a separate input captions selector for each Teletext page of your input. Specify the input Teletext page number for **Page number**.

For information about setting up the output of this captions workflow, see Teletext to Teletext, page remapping.

Teletext to other captions formats

When your input captions are Teletext and your output captions are another format, set up one input captions selector for each input Teletext page. Specify the input Teletext page number for **Page number**.

For information about setting up the output of this captions workflow, see the section on your output format in Setting up captions in outputs.

IMSC, SCC, SMPTE-TT, SRT, STL, TTML (sidecar) input captions

IMSC, SCC, SMPTE-TT, SRT, STL, and TTML are sidecar captions formats. With these formats, you provide input captions as a separate file. Depending on your output captions settings, AWS Elemental MediaConvert passes them through to the output in the same format or converts them into another sidecar format.

All sidecar captions

In all cases, create one captions selector for each input captions file.

In **Source file**, enter the URI to the captions input file that is stored in Amazon S3 or on an HTTP(S) server. For Amazon S3 inputs, you can specify the URI directly or choose **Browse** to select from your Amazon S3 buckets. For HTTP(S) inputs, provide the URL to your input video file. For more information, see HTTP input requirements.

IMSC captions

MediaConvert supports IMSC as an input captions format either as a sidecar file or as part of an IMF source. If your input IMSC captions are part of an IMF package, see IMSC input captions (as part of an IMF source). For restrictions on IMSC support, see IMSC captions support in AWS Elemental MediaConvert.

SMPTE-TT captions

You can use SMPTE-TT input captions that are text-only, that have captions images included in the captions file with base64 encoding (`smpte:image encoding="Base64"`), and that use external references to captions images (`smpte:backgroundImage`).

When your captions use external references to images, those images must be located in the same Amazon S3 bucket and folder as your captions file. For example, say this is the Amazon S3 path to your SMPTE_TT file: `DOC-EXAMPLE-BUCKET/mediaconvert-input/captions/my-captions-spanish.ttml`. Then you must store the image files that the captions file references here: `s3://DOC-EXAMPLE-BUCKET/mediaconvert-input/captions/`.

SRT captions

MediaConvert supports SRT input captions with UTF-8 character encoding.

Synchronizing sidecar captions and video

To make sure that your captions are properly synchronized with your video, check that the value for **Timecode source** in the **Video selector** section matches the timecodes in your captions file. For example, if your video has embedded timecodes starting at 01:00:00:00, but the timecodes in your captions file start at zero, change the default value for the video selector **Timecode source** from **Embedded** to **Start at 0**. If other aspects of your job prevent that, use the **Time delta** setting to adjust your captions, as described in Use cases for time delta.

> **Note**
>
> MediaConvert handles the alignment of captions with video differently depending on whether the caption format is timecode-based or timestamp-based. For more information, see Input timecode source and captions alignment.

Enter a positive or negative number in **Time delta** to modify the time values in the captions file. By default, time delta is measured in seconds. For example, enter **15** to add 15 seconds to all the time values in the captions file. Or, enter **-5** to subtract 5 seconds from the time values in the captions file. To specify in milliseconds instead, set **Time delta units** to **Milliseconds**.

If the value you enter for **Time delta** would result in captions before or after your video, those captions will not be present in your output.

> **Note**
>
> When converting from SCC to SRT, MediaConvert first rounds the value you set for **Time delta** to the nearest input frame. MediaConvert uses this rounded value when calculating output SRT timings.

Topics

Input timecode source and captions alignment

When you adjust your input timeline by setting the input **Timecode source** to **Start at 0** or **Specified start**, MediaConvert behaves as though your input has embedded timecodes that start when you specify. But MediaConvert doesn't change the timecodes or timestamps in your sidecar captions files. Therefore, the way that you align your captions depends on your captions format.

Timecode-based sidecar formats (SCC, STL)

Some captions formats, including SCC and STL, define where captions are placed in the video by timecode. With these formats, MediaConvert places each caption on the frames specified in the captions file, according to each frame's timecode in the input timeline. To adjust your captions to start at a different time than that, use the **Time delta** setting. For more information, see Use cases for time delta.

MediaConvert establishes the input timeline based on the value you choose for the input **Timecode source** setting.

For example, if your SCC file specifies that the first caption should appear at 00:05:23:00 and you set **Timecode source** to **Specified start** and **Start timecode** to 00:04:00:00, the first caption will appear in your output one minute and twenty-three seconds into the video. If you set **Timecode source** to **Specified start** and **Start timecode** to 01:00:00:00, you won't see captions when you expect, because 00:05:23:00 occurs before the start of your video, according to the input timeline.

Timestamp-based sidecar formats (SRT, SMI, TTML)

Some captions formats, including SRT, SMI, and TTML, allow for definition of where captions are placed in the video by timestamp. With these, MediaConvert measures the placement of the captions by the distance, in time, from the start of the video. This is true regardless of whether the captions file specifies placement with timecode or timestamp.

Therefore, your captions appear at the time specified in the captions file without regard to the video timecodes. For example, if your SRT file specifies that the first caption should appear at 00:05:23:00 or at 00:05:23,000 and you set **Timecode source** to **Specified start** and **Start timecode** to 00:04:00:00, the first caption will still appear in your output five minutes and twenty-three seconds into the video.

To adjust your captions to start at a different time than that, use the **Time delta** setting. For more information, see Use cases for time delta.

Formats that embed captions in the video stream (CEA/EIA-608, CEA/EIA-708)

Some captions formats embed the captions directly in the video frame or the video frame metadata. With these, MediaConvert keeps the captions with the frames that they are embedded in, regardless of the timecode settings.

Use cases for time delta

How you use **Time delta (TimeDelta)** depends on the problem you're trying to solve and the captions format that you're working with.

By default, you specify the time delta in seconds. If you want to specify it in milliseconds instead, set **Time delta units (TimeDeltaUnits)** to **Milliseconds (MILLISECONDS)**.

Adjusting for different timecodes between video and captions files

With timecode-based captions formats, such as SCC and STL, the timecodes in the captions might be relative to a starting timecode that is different from the starting timecode embedded in the video. You use **Time delta** to adjust for the difference.

Example problem: Your video file might have embedded timecodes that start at 00:05:00:00. The first instance of dialogue that requires captions might be one minute into the video, at timecode

00:06:00:00. Your captions file might be written on the assumption that your video timecodes start at 00:00:00:00, with the first caption starting at 00:01:00:00. If you don't use **Time delta**, MediaConvert would not include this first caption because it occurs before the start of the video.

Solution: Add five minutes to the captions. Enter `300` for **Time delta**.

Adjusting captions after synchronizing video and audio

Your timecode-based (SCC or STL) captions might be aligned with the timecodes that are embedded in your video, but you might need to use the input **Timecode source** setting to align your audio. This creates a difference between the video and captions, which you need to adjust for. You don't need to make this adjustment with timestamp-based captions formats, such as SRT, SMI, and TTML.

For more information about aligning captions when you use input **Timecode source**, see [Input timecode source and captions alignment].

Example problem: Your video file might have embedded timecodes that start at 00:05:00:00 and the first instance of dialogue that requires captions might be one minute into the video, at timecode 00:06:00:00. Your captions file is written to sync correctly, with the first caption starting at 00:06:00:00. But you need to change your embedded captions in your input to sync correctly with your audio file. So you set the input **Timecode source** to **Start at Zero**. If you don't use **Time delta**, MediaConvert would put the first caption in your output at six minutes into the video.

Solution: Subtract five minutes from the captions. Enter `-300` for **Time delta**.

Correcting slight errors in captions sync

With any type of sidecar format, there might be a small error in your input captions file, so that the captions are consistently a little late or a little early.

Example problem: Your video has embedded captions that start at zero. The first instance of dialogue that requires captions is at 00:06:15:00, but the captions appear on the screen three seconds late, at 00:06:18:00.

Solution: Subtract three seconds from the captions. Enter `-3` for **Time delta**.

Converting dual SCC input files to embedded captions

If you want to use two SCC files as your captions input and embed the captions as two output captions channels embedded in your output video stream, set up your captions according to this procedure.

To convert dual SCC to embedded captions

1. Set up two input captions selectors. Follow the procedure in Creating input captions selectors. Specify values as follows:
   - In each captions selector, choose **SCC** for **Source**.
   - For **Source file**, choose one of your input SCC files in each selector.
   - If you want both 608 and 708 captions embedded in your outputs, choose **Upconvert** for **Force 608 to 708 upconvert** in both captions selectors.
2. Set up captions in your outputs. Follow the procedure in Setting up captions in outputs. Follow these specific choices:
   - Specify the captions in the same output as the video that you want the captions embedded in.
   - Choose **Add captions** twice, to create **Captions 1** and **Captions 2** tabs in the **Encoding settings** section.
   - For **Captions source**, in each of the captions tabs, choose one of the captions selectors that you created in the preceding step of this procedure.
   - For **CC channel number**, choose a number for each of the captions tabs that don't share a field. For example, in **Captions 1**, choose **1** for **CC channel number** and in **Captions 2**, choose **3** for **CC channel number**.

     Don't choose the combinations 1 and 2 or 3 and 4, because those pairs of channels share the same field.
   - If you chose **Upconvert** in the preceding step of this procedure, optionally specify a service number for **708 service number**. Within an output, each captions tab must specify a different service number.

     If you upconvert and don't specify a value for **708 service number**, the service uses the value that you specify for **CC channel number** as your 708 service number.

TTML style formatting

AWS Elemental MediaConvert reads the style formatting of your input captions when your job runs. If you notice issues with the style formatting of your output, we recommend checking the formatting of your input captions or setting **Style passthrough** to Enabled. The following topics provide guidance for using fonts, heritable and non-heritable attributes, and right to left languages in your TTML input captions.

Specifying fonts

MediaConvert supports the following generic font families listed in the [TTML2 W3C recommendation](https://www.w3.org/TR/ttml2/):

- default
- monospace
- sansSerif
- serif
- monospaceSansSerif
- monospaceSerif
- proportionalSansSerif
- proportionalSerif

For the best results, specify a generic font family within your TTML input captions. If you specify an individual font instead, MediaConvert will map the font to one of the generic font families listed above.

Heritable and non-heritable attributes

Style attributes are either heritable or non-heritable. The [TTML 2 W3C recommendation](https://www.w3.org/TR/ttml2/) lists these under *inherited* for each style attribute.

Include non-heritable style attributes in every element that you want them to apply to.

For example, `tts:backgroundColor` is a non-heritable style attribute. The following results in *hello* with a red background color and *world* with no background color:

```
<span tts:backgroundColor="red">hello<br/>world</span>
```

You can fix the above formatting so that *hello world* both have a red background color by using individual spans, each with their own style attributes, like in this example:

```
<span><span tts:backgroundColor="red">hello</span> <br/> <span
tts:backgroundColor="red">world</span></span>
```

Right to left languages

MediaConvert supports both left to right and right to left text directions within TTML.

When you don't specify text direction, MediaConvert uses left to right.

To specify right to left, include a `tts:direction="rtl"` attribute. If your text has a mix of bidirectional characters, also include a `tts:unicodeBidi="embed"` attribute as described in the TTML2 W3C recommendation. Note that `tts:unicodeBidi` is a non-heritable attribute.

IMSC input captions (as part of an IMF source)

AWS Elemental MediaConvert supports IMSC as an input captions format either as a sidecar file or as part of an IMF source. If your input IMSC captions are in a sidecar file, see IMSC, SCC, SMPTE-TT, SRT, STL, TTML (sidecar) input captions.

When your input IMSC captions are part of an IMF source, you don't specify the source file for IMSC captions. That information is in the CPL file that you specify for your job input. For restrictions on IMSC support, see IMSC captions support in AWS Elemental MediaConvert.

Number of captions selectors for IMSC

Create one captions selector per track.

Track number

Specify which captions you want by providing a track number. The track numbers correspond to the order that the tracks appear in the CPL file. For example, if your CPL file lists your French captions first, set **Track number** to **1** to specify the French captions.

In your JSON job specification

If you use the API or an SDK, you can find these settings in the JSON file of your job. These settings are under `Inputs`, as in the following example:

```
"Inputs": [


      {
        ...

        "CaptionSelectors": {
          "Captions Selector 1": {
            "SourceSettings": {
              "SourceType": "IMSC",
```

```
            "TrackSourceSettings": {
              "TrackNumber": 1
            }
          }
        },

        "Captions Selector 2": {
          "SourceSettings": {
            "SourceType": "IMSC",
            "TrackSourceSettings": {
              "TrackNumber": 4
            }
          }
        },
        ...
```

WebVTT input captions (as part of an HLS source)

AWS Elemental MediaConvert supports WebVTT as an input captions format either as a sidecar file or as part of an HLS source. If your input WebVTT captions are in a sidecar file, see IMSC, SCC, SMPTE-TT, SRT, STL, TTML (sidecar) input captions.

When your input WebVTT captions are part of a HLS source, you don't need to specify the source WebVTT manifest file for the WebVTT captions. That information is in the main HLS input file that you specify in your job input. You will need to enable the **Use HLS Rendition Group** and use the following settings.

Number of captions selectors for WebVTT

Create one captions selector per WebVTT source.

Rendition Group Id

Specify which captions group you want by providing a group id. The group id corresponds to EXT-X-MEDIA, GROUP-ID tag in your HLS manifest. For example, if your HLS manifest file lists your French captions in a specific group "subs", set **Rendition Group ID** to **subs** to specify the French captions group id.

Rendition Name

Specify which captions group you want by providing a rendition name. The rendition name corresponds to EXT-X-MEDIA, NAME tag in your HLS manifest. For example, if your HLS manifest file lists your French captions in a rendition name called "French", set **Rendition Name** to **French** to specify the French captions rendition name.

Rendition Language Code

Specify which captions group you want by providing an ISO 639-3 language code. The language corresponds to EXT-X-MEDIA, LANGUAGE tag in your HLS manifest. For example, if your HLS manifest file lists your French captions in a language code of "FRA", set **Rendition Language Code** to **FRA** to specify the French captions rendition language code.

In your JSON job specification

If you use the API or an SDK, you can find these settings in the JSON file of your job. These settings are under `Inputs`, as in the following example:

```
"Inputs": [


      {
        ...

"CaptionSelectors": {
  "Caption Selector 1": {
    "SourceSettings": {
      "SourceType": "WebVTT",
      "WebvttHlsSourceSettings": {
        "RenditionGroupId": "subs",
        "RenditionName": "French",
        "RenditionLanguageCode": "FRA"
      }
    }
  }
}
          ...
```

Creating outputs

A single MediaConvert job can create outputs as a standalone file (for example, an .mp4 file), a set of files for adaptive bitrate (ABR) streaming (for example, an Apple HLS package), or combinations of both. When you create output groups and the outputs within them, you specify the number and types of files that your job generates.

When your MediaConvert job is complete, you can use Amazon CloudFront, or another content distribution network (CDN), to deliver your streaming package. The CDN gets your video to the people who want to view it. For more information, see Delivering video on demand (VOD) with CloudFront.

The topics in this section explain the relationship between MediaConvert output groups, MediaConvert outputs, and the actual output files that MediaConvert delivers to you.

Topics

Setting up captions in outputs

The location of the captions in a job depends on your output captions format: Your captions might be in the same output as your video, a separate output in the same output group as your video, or in an entirely separate output group. How you set up multiple captions tracks also depends on the output captions format.

For a full list of supported input and output captions, see Captions supported by AWS Elemental MediaConvert.

For information about how to set up captions in your input, see Setting up captions in AWS Elemental MediaConvert jobs.

The following procedure shows how to set up captions for different outputs.

To set up captions for different outputs

1. Open the MediaConvert console at https://console.aws.amazon.com/mediaconvert.
2. Choose **Create job**.

3. Set up your input, output groups, and outputs for video and audio, as described in Configuring jobs in MediaConvert and Creating outputs.
4. Create input captions selectors as described in the section called “Creating input captions selectors”.
5. Determine where in your job to specify the captions. This choice depends on the output captions format. Consult the relevant topic below to look this up.
6. In the left pane of the **Create job** page, choose the appropriate output from the list of outputs.
7. Under **Encoding settings**, choose **Add caption**. This displays a captions settings area under **Encoding settings**.
8. If your output captions format requires a separate group of captions settings for each track in the output, choose **Add captions** again until you have one captions group for each track. To determine whether you need one captions settings group for all tracks or one for each track, see the relevant topic below.
9. Under **Encoding settings**, choose **Captions 1** from the list.
10. Under **Captions source**, choose a captions selector. This selects the track or tracks that you associated with the selector when you set up your input, so that AWS Elemental MediaConvert will include those captions in this output.
11. Under **Destination type**, choose an output captions format. Check the section called “Supported captions workflows, input captions in the same file as video” to ensure that you are choosing a supported format.
12. Provide values for any additional fields as described in the relevant topic below.

Details by output captions format

- CEA/EIA-608 and CEA/EIA-708 (embedded) output captions
- DVB-Sub output captions
- IMSC, TTML, and WebVTT (sidecar) output captions
- SCC, SRT, and SMI (sidecar) output captions
- Teletext output captions
- Burn-in output captions
- Settings for accessibility captions

CEA/EIA-608 and CEA/EIA-708 (embedded) output captions

Where to specify the captions

Put your captions in the same output group and the same output as your video.

How to specify multiple captions tracks

- If your input captions format is embedded (that is, you are passing through embedded-to-embedded), you need to create only one group of captions settings. The captions selector that you choose under **Captions source** includes all tracks from the input.
- If your input captions are two SCC files, you can create output captions as two output captions channels that are embedded in your output video stream. For more information, see Converting dual SCC input files to embedded captions.
- If your input captions are not embedded or SCC, you can include only one captions track per output. In each output, include one group of captions settings. Under **Captions source**, choose the selector that is set up for the track that you want to include.

Embedded and ancillary captions in MXF outputs

Whether your MXF output can contain ancillary captions depends on the MXF profile:

- MXF XDCAM HD: This MXF profile specifies ancillary data in the smpte 436 track. With these outputs, MediaConvert copies your embedded captions to the smpte 436 ancillary track in addition to including it in the video stream.
- MXF D-10: This MXF profile specification doesn't allow for ancillary data. Therefore, your MXF D-10 outputs only have captions embedded in the video stream.

MediaConvert determines an output's MXF profile based on the values for the following encoding settings:

- Resolution
- Frame rate
- Video codec profile
- Interlace mode

For information about which values for these settings are valid for which MXF profile, see the relevant specifications. For XDCAM HD, see RDD 9:2009 - SMPTE Standard Doc - MXF Interoperability Specification of Sony MPEG Long GOP Products in the IEEE Xplore Digital Library. For MXF D-10, see ST 356:2001 - SMPTE Standard - For Television — Type D-10 Stream Specifications — MPEG-2 4:2:2P @ ML for 525/60 and 625/50.

DVB-Sub output captions

Where to specify the captions

Put your captions in the same output group and the same output as your video.

How to specify multiple captions tracks

- If your input captions are the same format as your output captions (passthrough), you need to create only one group of captions settings. The captions selector that you choose under **Captions source** includes all tracks from the input.
- If your input captions are in a different format, create one group of captions settings for each track. Put each group of captions settings in the same output. They will appear in the list of settings groups as Captions 1, Captions 2, and so forth. In each group of settings, choose the captions selector under **Captions source** that is set up for the track that you want to include.

How to specify the font script

AWS Elemental MediaConvert automatically selects the appropriate script for your captions, based on the language that you specify in the output captions settings. If the language that you choose has more than one possible script, specify the script that you want.

To ensure that the service uses the correct font script

1. In the **Captions** section under **Encoding settings**, for **Language**, choose the language of the captions text.
2. If the language that you specify has more than one possible script, use **Font script** to specify the script.

   For example, if you choose **Chinese** (ZH) for **Language**, use **Font script** to choose either **Simplified Chinese** or **Traditional Chinese**. In this case, if you don't specify a value for **Font script**, the service defaults to simplified Chinese.

> **Tip**
>
> In most cases, for **Font script** you can keep the default value of **Automatic**. When you do, the service chooses the script based on the language of the captions text.

IMSC, TTML, and WebVTT (sidecar) output captions

If your output captions are IMSC, TTML or WebVTT format, set them up in your outputs according to the following information. For restrictions on IMSC support, see IMSC captions support in AWS Elemental MediaConvert.

Where to specify the captions

Put your captions in the same output group, but a different output from your video.

After you add captions to an output, delete the **Video** and **Audio 1** groups of settings that the service automatically created with the output.

To delete the Video and Audio 1 groups of settings

1. On the **Create job** page, in the **Job** pane on the left, under **Output groups**, choose the output that contains the groups of settings that you want to delete.
2. The **Video** group of settings is automatically displayed in the **Stream settings** section. Choose the **Remove video selector** button.
3. The **Audio 1** group of settings is automatically displayed in the **Stream settings** section. Choose the **Remove** button.

How to specify multiple captions tracks

Put each captions track in its own output.

> **Note**
>
> The captions track that you specify first in your job is signaled as the default track in the HLS manifest.

Sidecar captions container options

Depending on your output group, you can choose the captions container for IMSC and TTML captions outputs.

For **DASH ISO** output groups, you can choose from these:

- Fragmented MP4 (`.fmp4`)
- Raw (`.xml` for IMSC, `.ttml` for TTML)

For all other output groups, IMSC and TTML files are raw.

To specify the captions container for IMSC and TTML captions in DASH ISO output groups

1. Set up the outputs in your **DASH ISO** output group as described in Creating outputs in ABR streaming output groups. Put captions in a separate output.
2. On the **Create job** page, in the **Job** pane on the left, choose the captions output.
3. In the **Output settings** section on the right, choose **Container settings**, and then enable **DASH container settings**.
4. For **Captions container**, keep the default **Raw** or choose **Fragmented MPEG-4**.

SCC, SRT, and SMI (sidecar) output captions

Where to specify the captions

Put your captions in the same output group, but a different output from your video.

After you add captions to an output, delete the **Video** and **Audio 1** groups of settings that the service automatically created with the output.

To delete the Video and Audio 1 groups of settings

1. On the **Create job** page, in the **Job** pane on the left, under **Output groups**, choose the output that contains the groups of settings that you want to delete.
2. The **Video** group of settings is automatically displayed in the **Stream settings** section. Choose the **Remove video selector** button.
3. The **Audio 1** group of settings is automatically displayed in the **Stream settings** section. Choose the **Remove** button.

How to specify multiple captions tracks

For each SRT, SCC or SMI output you must have one output per caption selector. In the caption output, choose the captions selector under **Captions source** that is set up for the track that you want to include. They will appear in the list of settings groups as **Captions Selector 1**, **Captions Selector 2**, and so forth.

Teletext output captions

How you set up your output Teletext captions depends on whether you want to move the captions to different Teletext pages or to just pass through your captions exactly from the input to the output.

Teletext to Teletext passthrough

When your input captions format is Teletext and you want your output captions to be on the same pages, with the same styling, as the input, then you can pass through the input captions to your output. To do so, set up your captions this way:

- Make sure that your input captions are set up with one captions selector. For more information, see Teletext input captions.
- In the same output group and same output as your video, create one captions tab. This one captions tab represents all of your output captions, regardless of the number of output Teletext pages you have.
- In your output captions tab, choose your input captions selector for **Captions source**.
- Don't specify values for any other settings on the output captions tab.

When you work directly in your JSON job specification, one captions tab corresponds to one child of `CaptionDescriptions`.

Teletext to Teletext, page remapping

When your input captions format is Teletext and, in your output, you want to change the Teletext pages that your captions are on, you specify the pages in the input and output. To do so, set up your captions this way:

- Make sure that your input captions are set up with one captions selector for each Teletext page and that you specify the page number in the settings for each input captions selector. For more information, see Teletext input captions.

- In the same output group and same output as your video, create one captions tab for each output Teletext page.
- In each output captions tab, choose one of your input captions selectors for **Captions source**.
- In each output captions tab, for **Page number**, specify the Teletext page number that you want for those captions in your output. Optionally, provide values for **Language**, **Description**, and **Page types**.

Teletext from other captions formats

When your input captions are in a format other than Teletext, you must specify the Teletext pages for your output captions. MediaConvert supports these captions workflows:

- A single input captions track to a single output Teletext page.
- A single input captions track to multiple output Teletext pages. Each output page duplicates the contents of the others.
- Multiple input captions tracks to multiple output Teletext pages. You use captions selectors to specify which captions to include on each output Teletext page.

Set up your captions like this:

- Make sure that your input captions are set up with one captions selector for each captions track that you intend to map to a Teletext page. For more information, see Creating input captions selectors.
- In the same output group and same output as your video, create one captions tab for each output Teletext page.
- In each output captions tab, choose one of your input captions selectors for **Captions source**.
- In each output captions tab, for **Page number**, specify the Teletext page number that you want for those captions in your output. Optionally, provide values for **Language** and **Description**.

Burn-in output captions

Burn-in is a way to deliver captions, rather than a captions format. Burn-in writes the captions directly on your video frames, replacing pixels of video content with the captions. If you want burn-in captions in an output, set up the captions according to the following information.

Where to specify the captions

Put your captions in the same output group and the same output as your video.

How to specify multiple captions tracks

You can burn in only one track of captions in each output.

How to use style passthrough

You can choose how to stylize the burn-in caption text that appears in your output video. There are a few options, including style passthrough, default settings, or manual overrides.

When you set Style passthrough to Enabled, MediaConvert uses the available style and position information from your input captions. Note that MediaConvert uses default settings for any missing style information.

MediaConvert supports style passthrough for the following input caption formats:

- Ancillary
- Embedded
- SCTE-20
- SCC
- TTML
- STL (EBU STL)
- SMPTE-TT (text based)
- Teletext
- IMSC
- WebVTT

When you set Style passthrough to Disabled, MediaConvert ignores style information from your input and uses default settings: white text with black outlining, bottom-center positioning, and automatic sizing.

Whether you set style passthrough to enabled or not, you can also choose to manually override any of the individual style options.

> **Note**
>
> TTML and TTML-like (IMSC, SMPTE-TT) inputs have special style formatting requirements. For more information, see TTML style formatting.

How to specify the font script

AWS Elemental MediaConvert automatically selects the appropriate script for your captions, based on the language that you specify in the output captions settings. If the language that you choose has more than one possible script, specify the script that you want.

To ensure that the service uses the correct font script

1. In the **Captions** section under **Encoding settings**, for **Language**, choose the language of the captions text.
2. If the language that you specify has more than one possible script, use **Font script** to specify the script.

   For example, if you choose **Chinese** (ZH) for **Language**, use **Font script** to choose either **Simplified Chinese** or **Traditional Chinese**. In this case, if you don't specify a value for **Font script**, the service defaults to simplified Chinese.

   > **Tip**
   >
   > In most cases, for **Font script** you can keep the default value of **Automatic**. When you do, the service chooses the script based on the language of the captions text.

Non-english fonts and unsupported characters

When your input font uses a non-English font script, your output burn-in captions may contain unsupported Unicode characters #. To resolve, set **Style passthrough** to **Enabled**.

Settings for accessibility captions

You can mark closed caption tracks as an accessibility aid for those who are hearing impaired. The following section describes how accessibility captions work in AWS Elemental MediaConvert.

HLS and CMAF

When you output HLS or CMAF and include an ISMC or WebVTT captions track, you can add accessibility attributes for captions to your output manifest. MediaConvert adds these attributes according to sections 4.5 and 4.6 of the [HLS authoring specification for Apple devices](https://developer.apple.com/documentation/http_live_streaming/hls_authoring_specification_for_apple_devices).

When you set **Accessibility subtitles** (`accessibility`) to **Enabled** (`ENABLED`), MediaConvert adds the following attributes to the captions track in the manifest under `EXT-X-MEDIA`: `CHARACTERISTICS="public.accessibility.describes-spoken-dialog,public.accessibility.describes-music-and-sound"` and `AUTOSELECT="YES"`.

Keep the default value, **Disabled** (`DISABLED`), if the captions track is not intended to provide such accessibility. MediaConvert will not add the attributes from the previous paragraph.

Using output groups to specify a streaming package type or standalone file

AWS Elemental MediaConvert output functions differ based on which type of output group it is a part of.

File

In a **File** output group, each output that you set up results in a standalone output file.

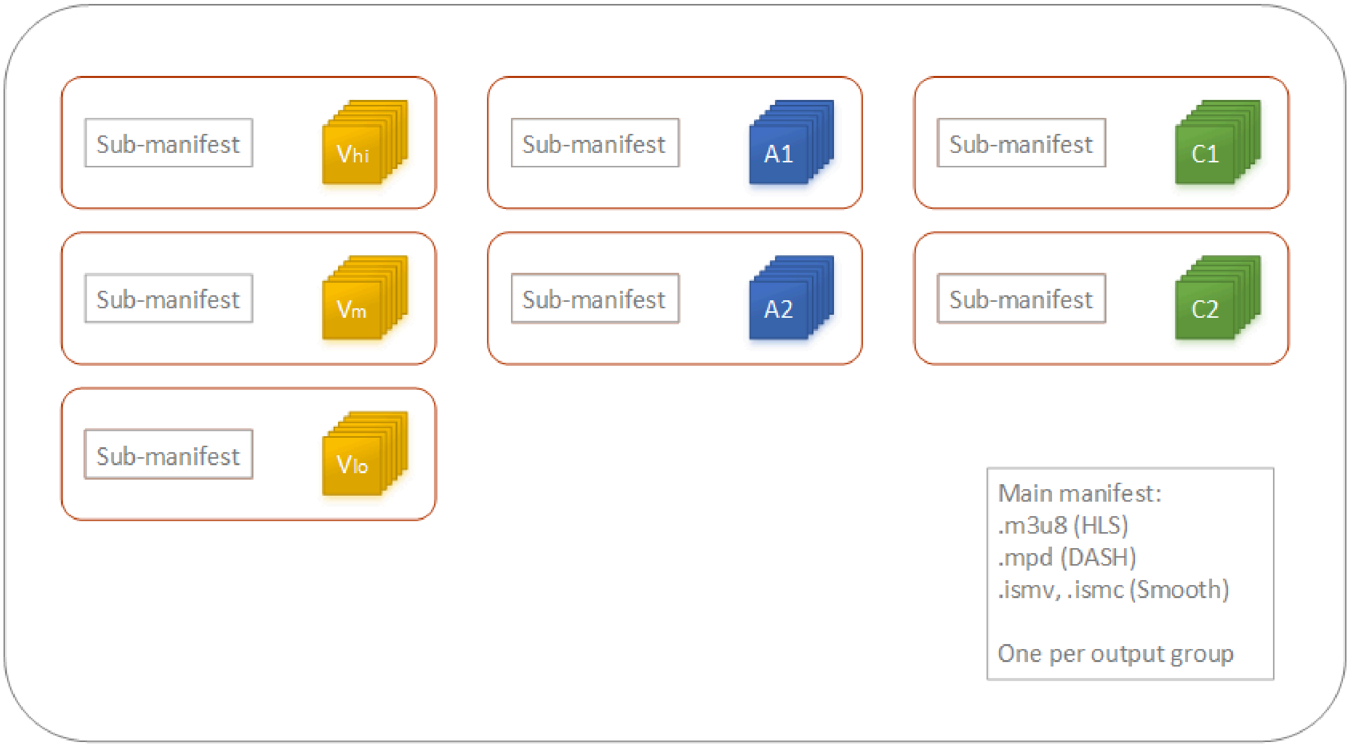
For example, you might set up one output that contains all the video, audio, and captions together. You might also set up a separate output for sidecar captions, such as TTML.

Streaming output packages

In the following output groups, the outputs that you set up are separate parts of a single adaptive bitrate (ABR) streaming package: CMAF, Apple HLS, DASH ISO, and Microsoft Smooth Streaming.

In an ABR output group, each output is usually one element of the media. That is, each output is one slice in the adaptive bitrate (ABR) stack. For example, you might have an output for each of three resolutions of video, an output for each of two audio language tracks, and an output for each of two captions languages.

The following illustration shows the relationship between outputs in an ABR output group and the files that MediaConvert creates. Each orange box corresponds to an output within the output group. In this example, there are three resolutions of video, audio in two languages, and captions in two languages. The package contains segmented audio, video, and captions files, plus manifest files that tell the player which files to download and when to play them.



A single job can generate zero to many standalone files and zero to many streaming packages. To create more than one standalone file, add a single file output group to your job and add multiple outputs to that output group. To create more than one streaming package, add multiple **CMAF**, **AppleHLS**, **DASH ISO**, or **Microsoft Smooth Streaming** output groups to your job.

The following illustration shows a MediaConvert job that generates two standalone .mp4 files, two Apple HLS packages, and a CMAF package. A single file output group with two outputs results in two standalone files. A single Apple HLS output group with seven outputs results in a single viewable package with seven ABR slices.

For information about setting up output groups and outputs within your job, see [Configuring jobs in MediaConvert](#).

Choosing your ABR streaming output groups

To create media assets for people to stream to their devices, choose one or more of the adaptive bitrate (ABR) output groups: Apple HLS, DASH ISO, Microsoft Smooth Streaming, or CMAF. The type of output group determines which media players can play the files that MediaConvert creates from that output group.

Note

When you set up CMAF, DASH ISO, or Microsoft Smooth Streaming output groups, make sure to set your fragment length correctly. For information about setting fragment length, see Setting the fragment length for streaming outputs.

The following table summarizes the relationships between output groups and media players.

Media players	Use this output group
Apple devices, earlier than approximately 2013	Apple HLS
Apple devices, newer	CMAF
Android devices, most smart TVs	CMAF or DASH ISO
Microsoft devices	Microsoft Smooth Streaming

Note

MediaConvert bills for each minute of transcoded output time, not per job. Therefore, when you add output groups to a job, it becomes more expensive.
For example, a job with an Apple HLS package and a DASH ISO package costs twice as much as a job with only one of those packages. This is assuming that the transcoding settings are the same.

To determine which output groups you need

1. Decide which devices you want end viewers to be able to play the transcoded media asset on. If you want your asset to play on every possible device, include these output groups:

   - Apple HLS
   - DASH ISO or CMAF
   - Microsoft Smooth Streaming

2. Consider whether to use advanced encoding features. To deliver either of the following to Apple devices, you must also include a CMAF output group:

    - High-dynamic-range (HDR) video
    - H.265 (HEVC) encoded video

    If you include a CMAF output, you don't need to create a DASH ISO output because all the common DASH-compatible players are also CMAF-compatible.

    > **Note**
    >
    > There are a few uncommon DASH players that explicitly require the video segmentation extension type .mp4. MediaConvert outputs CMAF video segments in the .cmfv format. To create output that is compatible with these players, include a DASH ISO output group in your job.

3. Consider cost trade-off.

    If you don't need to support players produced earlier than approximately 2013, and if you don't need to support the rare DASH players that require .mp4 video segments, you can include a single CMAF output group instead of both DASH ISO and Apple HLS. Creating a single CMAF package instead of separate DASH ISO and Apple HLS packages can also offer cost savings in your video storage and distribution. This is because you must store and distribute only one set of video and audio files.

Setting the fragment length for streaming outputs

For all ABR streaming output groups other than HLS (CMAF, DASH, and Microsoft Smooth Streaming), the value that you specify for **Fragment length** (`FragmentLength`) must work with the other output settings that you specify. If you set **Fragment length** incorrectly, when viewers watch the output video, their player might crash. This can happen because the player expects additional segments at the end of the video and requests segments that don't exist.

Fragment length is constrained by your values for **Closed GOP cadence** (`GopClosedCadence`), **GOP size** (`GopSize`), and **Frame rate** (`FramerateNumerator`, `FramerateDenominator`). For information about finding these settings on the console and in your JSON job specification, see Finding the settings related to fragment length.

Note

When you set your output **Frame rate** to **Follow source**, make sure that the frame rate of your input video file works with the value that you specify for the output **Fragment length**. The frame rate of your input video file functions as your output frame rate.

Topics

Rule for fragment length

Fragment length must be a whole number and must be a multiple of this value: **GOP size** x **Closed GOP cadence** ÷ **Frame rate**

Fragment length examples

Example: Correct settings

Closed GOP cadence = 1

Frame rate = 30

GOP size = 60 frames

Fragment length = 2

Example: Incorrect settings

Closed GOP Cadence = 1

Frame rate = 50

GOP size = 90 frames

Fragment length = 2

Finding the settings related to fragment length

When you set **Fragment length**, check your values for **Closed GOP cadence**, **GOP size**, and **Frame rate**.

Fragment length

You can set the fragment length using either the console or the JSON job specification. The **Fragment length** setting applies to an output group and affects every output in the group.

To find the Fragment length setting (console)

1. On the **Create job** page, in the **Job** pane on the left, under **Output groups**, choose the name of your CMAF, DASH ISO, or Microsoft Smooth Streaming output group.
2. In the group settings section on the right, find **Fragment length**.

   The group settings section is titled **CMAF group settings**, **DASH ISO group settings**, or **MS Smooth group settings**.

To find the Fragment length setting (JSON job specification)

- Find `FragmentLength` as a child of `OutputGroupSettings`, as in the following example.

```
{
  "Settings": {
    ...
    "Inputs": [
      ...
    ],
    "OutputGroups": [
      {
        "Name": "DASH ISO",
        "OutputGroupSettings": {
          "Type": "DASH_ISO_GROUP_SETTINGS",
          "DashIsoGroupSettings": {
            "SegmentLength": 30,
            "FragmentLength": 2,
            "SegmentControl": "SINGLE_FILE",
            "HbbtvCompliance": "NONE"
          }
        },
```

```
    ...
```

Closed GOP cadence, GOP size, and frame rate

You can set **Closed GOP cadence**, **GOP size**, and **Frame rate** using either the console or the JSON job specification. These settings apply to each output individually. Make sure that the values that you set for each output in the output group work with the value that you specify for the output group's **Fragment length**.

> **Note**
>
> Your ABR stack has multiple outputs. Make sure to set these values in each output.

To find the encoding settings for an output (console)

1. On the **Create job** page, in the **Job** pane on the left, under **Output groups**, choose the name of your output, such as **Output 1**, **Output 2**, and so on.
2. In the **Encoding settings** section, the **Video** tab is selected automatically. Find **Closed GOP cadence**, **GOP size**, and **Frame rate** on this tab.

To find the encoding settings for an output (JSON job specification)

- Find `GopClosedCadence`, `GopSize`, `FramerateNumerator`, and `FramerateDenominator` as children of the codec settings, as in the following example. In this example, the codec is H_264, so the parent of the codec settings is `H264Settings`.

```
{
  "Settings": {
    ...
    "Inputs": [
      ...
    ],
    "OutputGroups": [
      {
        "Name": "DASH ISO",
        ...
        },
        "Outputs": [
```

```
            {
              "VideoDescription": {
                ...
                "CodecSettings": {
                  "Codec": "H_264",
                  "H264Settings": {
                    "InterlaceMode": "PROGRESSIVE",
                    "NumberReferenceFrames": 3,
                    "Syntax": "DEFAULT",
                    "Softness": 0,
                    "GopClosedCadence": 1,
                    "GopSize": 60,
        ...
                    "FramerateNumerator": 60,
                    "FramerateDenominator": 1
                  }
                },
                ...
              },
```

HLS player version support

AWS Elemental MediaConvert automatically sets the player version metadata based on the features that you enable. Most HLS assets that you create with MediaConvert are compatible with HLS players version 2 and later.

This list shows the features that might require updated player support:

Add I-frame only manifest: HLS Output group > Output > Advanced > Add I-frame only manifest

When you choose **Include**, viewers can play the asset with HLS players version 4 and later.

When you choose **Exclude**, viewers can play the asset with HLS players version 2 and later.

Audio track type: HLS Output group > Output > Output settings > Advanced > Audio track type

When you choose one of the **Alternate audio** options for any audio variants, viewers can play the asset with HLS players version 4 and later.

When you choose **Audio-only variant stream** for **Audio track type**, or keep **Audio track type** unselected for all of your audio variants, viewers can play the asset with HLS players version 2 and later.

DRM encryption method: HLS output group > DRM encryption > Encryption method

When you choose **SAMPLE-AES** for **DRM encryption**, **Encryption method**, viewers can play the asset with HLS players version 5 and later.

When you choose any other value for **DRM encryption**, **Encryption method**, viewers can play the asset with HLS players version 2 and later.

Descriptive video service flag: HLS output group > Output (must be audio-only) > Output settings > Descriptive video service flag

This setting is also available in CMAF output groups: CMAF output group > Output > CMAF container settings > Advanced > Descriptive video service flag

To find this setting, your HLS or CMAF output must have only audio settings. In HLS outputs, you must remove the default **Video** tab.

When you choose **Flag** for **Descriptive video service flag**, viewers can play the asset with HLS players version 5 and later.

To create a compliant Apple HLS output: When you set **Descriptive video service flag** to **Flag**, you must also set **Audio track type** to **Alternative audio, auto select, default** or **Alternative audio, auto select, not default**.

Manifest duration format: HLS output group > Apple HLS group settings > Advanced > Manifest duration format

When you set your manifest duration format to **Integer**, viewers can play the asset with HLS players version 2 and later.

When you set your manifest duration format to **Floating point**, viewers can play the asset with HLS players version 3 and later.

Segment control: HLS output group > Apple HLS group settings > Segment control

When you set segment control to **Single file**, viewers can play the asset with HLS players version 4 and later.

When you set segment control to **Segmented files**, viewers can play the asset with HLS players version 2 and later.

Recommended encoding settings for video quality

When you create a job with AWS Elemental MediaConvert, the encoding settings that you choose affect video quality, file size, and player compatibility.

You can configure your job to allow MediaConvert to automatically select the best encoding settings for video quality, with a balanced output file size. Or, you can manually specify encoding settings to match your output or delivery requirements.

This section introduces basic concepts, describes typical settings, and provides guidance for choosing settings optimized for video quality.

Topics

Reference for GOP structure and frame types

When you create a job, the group of pictures (GOP) settings that you choose for your output affect video quality and player compatibility. This section introduces basic GOP concepts, describes typical GOP settings, and provides guidance for choosing settings optimized for video quality.

A GOP is a specific arrangement of compressed video frame types. These frame types include the following:

I-Frames

Intra-coded frames. Contain all of the information that a decoder uses decode the frame. Typically, I-frames use the most number of bits within a video stream.

IDR-Frames

Instantaneous Decoder Refresh frames. Similar to I-frames, they contain all of the information that a decoder uses to decode the frame. However, frames cannot reference any frame that comes before an IDR-frame.

P-Frames

Predicted frames. Contain the differences between the current frame and one or more frames before it. P-frames offer much better compression than I-frames and use fewer bits within a video stream.

B-Frames

Bidirectional predicted frames. Contain the differences between the current frame and one or more frames before or after it. B-frames offer the highest compression and take up the fewest bits within a video stream.

A typical GOP starts with an IDR-frame and follows with a repeating pattern of B- and P-frames. For example: `IDRBBPBBPBBPBB`

The following topics provide more information about individual GOP settings and recommend settings that are optimized for video quality.

GOP size recommended setting

GOP size is the number of frames in a GOP, and it defines the interval between IDR-frames. For example, if a GOP starts with an IDR-frame and has a combination of 29 B and P-frames, the GOP size is 30 frames.

A typical GOP size is 1–2 seconds long and corresponds to the video frame rate. For example, if your output frame rate is 30 frames per second, a typical GOP size is 30 or 60 frames.

When you set your output video codec to `AVC (H.264)` or `HEVC (H.265)`, set **GOP mode control** to `Auto`. This allows MediaConvert to select an optimal GOP size.

> **Note**
>
> Streaming video formats, including HLS, DASH, CMAF, and MSS, require the fragment or segment length to be a multiple of the GOP size. For more information, see Setting the fragment length for streaming outputs. When you set GOP mode control to Auto for these

video formats, MediaConvert automatically selects a compatible and optimized GOP size that's relative to the fragment or segment length.

B-frames between reference frames recommended setting

Defines the maximum number of B-frames that MediaConvert can use between reference frames.

A typical value is 1 or 2 if **GOP reference B-Frames** is set to `Disabled`, and 3–5 if **GOP reference B-frames** is set to `Enabled`.

When you set your output video codec to `AVC (H.264)` or `HEVC (H.265)`, keep **B-frames between reference frames** blank. This allows MediaConvert to select an optimal number of B-frames between reference frames.

Closed GOP cadence recommended setting

Closed GOP cadence defines the number of GOPs a P- or B-frame is able to reference across. A GOP can either be *open* or *closed*. Open GOPs can have frames that reference a frame from a different GOP, while closed GOPs have frames that reference only within the GOP itself.

When you set your output video codec to `AVC (H.264)` or `HEVC (H.265)`, keep **Closed GOP cadence** blank to allow MediaConvert to select an optimal closed GOP cadence.

Dynamic sub-GOP recommended setting

A dynamic sub-GOP can improve the subjective video quality of high-motion content. It does this by allowing the number of B-frames to vary.

When you set your output video codec to `AVC (H.264)` or `HEVC (H.265)`, set **Dynamic sub-GOP** to `Adaptive`. This allows MediaConvert to determine an optimal sub-GOP.

GOP reference B-frames recommended setting

When you set your output video codec to `AVC (H.264)` or `HEVC (H.265)`, set **GOP reference B-frames** to `Enabled` to allow B-frames to be referenced by other frame types. This improves the video quality of your output relative to its bitrate.

Min I-Interval recommended setting

Min I-Interval enforces a minimum number of frames between IDR-frames. This includes frames that are created at the beginning of a GOP or by scene change detection. Use Min I-Interval to improve video compression by varying GOP size when two IDR-frames would be created near each other.

When you set your output video codec to `AVC (H.264)` or `HEVC (H.265)`, keep **Min I-Interval** blank. This allows MediaConvert to select an optimal minimum I-interval.

Adaptive quantization recommended setting

Adaptive quantization selects the strength applied to the different quantization modes that MediaConvert uses, including flicker, spatial, and temporal quantization. MediaConvert uses adaptive quantization to assign bits according to the complexity of your video.

When you set your output video codec to `AVC (H.264)`, `HEVC (H.265)`, or `XAVC`, set **Adaptive quantization** to `Auto` to allow MediaConvert to select an optimal adaptive quantization.

Using variables in your job settings

You can use variables, also called *format identifiers*, in your job settings. Format identifiers are values that you can put in your job settings that resolve differently in your outputs depending on the characteristics of the input files or the job. They are particularly useful in output presets, job templates, and jobs that you intend to duplicate and re-use.

For example, you might use the date format identifier `$d$` for your **Destination** setting. If you want your outputs organized by the date and time that the job starts, for **Destination** you might enter **`s3://DOC-EXAMPLE-BUCKET1/$d$/`**. For a job that starts June 4, 2020, the service will create your outputs in `s3://DOC-EXAMPLE-BUCKET1/20200604/`.

For a list of the available format identifiers and examples of how to use them, see the section called "List of settings variables with examples".

For information about format identifiers that function differently in streaming outputs, see the section called "Using settings variables with streaming outputs".

Topics

List of settings variables with examples

The following table provides information about each of the format identifiers that you can use in your AWS Elemental MediaConvert job. For information about format identifiers that function differently in streaming outputs, see the section called "Using settings variables with streaming outputs".

Format identifier	Value to put in the job setting	Compatible job settings	Description and example
Date and time	`$dt$`	Destination Name modifier Segment modifier	UTC date and time of the start time of the job. Format: YYYYMMDDT HHMMSS Example: For a job that starts at 3:05:28 PM on June 4, 2020, **`$dt$`** resolves to `20200604T150528` .
Date	`$d$`	Destination Name modifier Segment modifier	UTC date of the start time of the job. Format: YYYYMMDD Example: For a job that starts on June 4, 2020, **`$d$`** resolves to `20200604`.
Time	`$t$`	Destination Name modifier Segment modifier	UTC start time of the job. Format: HHMMSS Example: For a job that starts at 3:05:28

Format identifier	Value to put in the job setting	Compatible job settings	Description and example
			PM, **`$t$`** resolves to `150528`.
Video bitrate	`$rv$`	Name modifier Segment modifier	The video bitrate of the output, in kilobits. For QVBR outputs, the service uses video max bitrate, in kilobits. Example: If you set **Encoding settings**, **Video**, **Bitrate (bits/s)** to **`50000000`**, **`$rv$`** resolves to `50000`.
Audio bitrate	`$ra$`	Name modifier Segment modifier	Total of all the audio bitrates in the output, in kilobits. Example: If you have an output with a single audio tab and you set **Encoding settings**, **Audio 1**, **Bitrate (kbit/s)** to **`256000`**, **`$ra$`** resolves to `256000`.

Format identifier	Value to put in the job setting	Compatible job settings	Description and example
Container bitrate	`$rc$`	Name modifier Segment modifier	Combined audio and video bitrate for the output, in kilobits. Example: You have an output with a **Video** settings tab and **Audio 1** settings tab. If you set **Encoding settings**, **Video**, **Bitrate (bits/s)** to **`5000000`** and you set **Encoding settings**, **Audio**, **Bitrate (bits/s)** to **`96000`** (96 kilobits), **`$rc$`** resolves to `5096`.
Video frame width	`$w$`	Name modifier Segment modifier	The frame width, or horizontal resolution, in pixels. Example: If you set **Encoding settings**, **Video**, **Resolution (w x h)** to **`1280`** x **`720`**, **`$w$`** resolves to `1280`.

Format identifier	Value to put in the job setting	Compatible job settings	Description and example
Video frame height	`$h$`	Name modifier Segment modifier	The frame height, or vertical resolution, in pixels. Example: If you set **Encoding settings**, **Video**, **Resolution (w x h)** to **`1280`** x **`720`**, **`$h$`** resolves to `720`.
Framerate	`$f$`	Name modifier Segment modifier	Framerate, in frames per second, truncated to the nearest whole number. Example: If your framerate is **`59.940`**, **`$f$`** resolves to `59`.
Input file name	`$fn$`	Destination Name modifier Segment modifier	Name of the input file, without the file extension. For jobs that have multiple inputs, this is the first file specified in the job. Example: If **Input 1** for your job is **`s3://DOC-EXAMPLE-BUCKET/my-video.mov`**, **`$fn$`** resolves to `my-video`.

Format identifier	Value to put in the job setting	Compatible job settings	Description and example
Output container file extension	`$ex$`	Name modifier Segment modifier	Varies depending on the output group. For **File group** outputs, this is the extension of the output container file. For other output groups, this is the extension of the manifest. Example for file group: If you choose **MPEG2-TS** for **Output settings**, **Container**, **`$ex$`** resolves to `m2ts`. Example for HLS group: If your output group is HLS, **`$ex$`** resolves to `m3u8`.

Format identifier	Value to put in the job setting	Compatible job settings	Description and example
$	$$	Name modifier Segment modifier	Escaped $. Example: Suppose that you provide the following values: • Input file name: **`file1.mp4`** • Destination: **`s3://DOC-EXAMPLE-BUCKET/`** • Name modifier: **`my-video$$hi-res-`** Your output file name and path resolves to `s3://DOC-EXAMPLE-BUCKET/my-video$hi-res-file1.mp4` .

Using settings variables with streaming outputs

Variables in your job settings, also called *format identifiers*, function differently for outputs in Apple HLS and DASH ISO output groups. Here are the differences:

For Apple HLS Outputs

When you use the date and time format identifiers (`$dt$`, `$t$`, `$d$`) in the **Segment modifier** setting, these format identifiers resolve to the completion time of each segment, rather than to the start time of the job.

> **Note**
>
> For jobs that use accelerated transcoding, segments might complete at the same time. This means that date and time format identifiers don't always resolve to unique values.

For DASH ISO Outputs

You can use two additional format identifiers in the **Name modifier** setting. These affect the DASH manifest in addition to the output file name. Here are the identifiers:

$Number$

In your output file names, `$Number$` resolves to a series of numbers that increment from 1. This replaces the default, nine-digit segment numbering in the segment file names. For example:

- If you specify **`video_$Number$`** for **Name modifier**, the service creates segment files named `video_1.mp4`, `video_2.mp4`, and so on.
- If you specify only **`video_`** for **Name modifier**, the service creates segment files named `video_000000001.mp4`, `video_000000002.mp4`, and so on.

In your DASH manifest, AWS Elemental MediaConvert includes `duration` and `startNumber` inside the `SegmentTemplate` element, like this: `<SegmentTemplate timescale="90000" media="main_video_$Number$.mp4" initialization="main_video_$Number$init.mp4" duration="3375000"/>`

> **Note**
>
> If you use the `$Number$` format identifier in an output, you must also use it in every other output of the output group.

$Bandwidth$

In your output file names, `$Bandwidth$` resolves to the value of **Video**, **Bitrate** plus the value of **Audio**, **Bitrate** in the output. Regardless of whether you include this format identifier, the service uses nine-digit segment numbering in the segment file names.

For example, suppose you specify these values:

- **Video**, **Bitrate (bits/s)**: **`50000000`**
- **Audio**, **Bitrate (kbits/s)**: **`96.0`** (96,000 bits/s)
- **Name modifier**: **`video_$Bandwidth$`**

The value for $Bandwidth$ resolves to 50,096,000. The service creates segment files named `video_50096000_000000001.mp4`, `video_50096000_000000002.mp4`, and so on.

In the manifest, AWS Elemental MediaConvert includes `duration` and `startNumber` inside the `SegmentTemplate` element, like this: `<SegmentTemplate timescale="90000" media="main_video_$Bandwidth$.mp4" initialization="main_video_$Bandwidth$init.mp4" duration="3375000"/>`.

$Time$

In your output file names, `$Time$` resolves to the duration, in milliseconds, of the segment. When you include this format identifier, the service doesn't use the default nine-digit segment numbering in the segment file names.

For example, if you specify **`video180__$Time$`** for **Name modifier**, the service creates segment files named `video180__345600.mp4`, `video180__331680.mp4`, and so on. In these examples, the segment durations are 345,600 ms and 331,680 ms.

In the manifest, AWS Elemental MediaConvert includes `SegmentTimeline` inside the `SegmentTemplate` element, like this:

```
<Representation id="5" width="320" height="180" bandwidth="200000"
 codecs="avc1.4d400c">
        <SegmentTemplate media="video180_$Time$.mp4"
 initialization="videovideo180_init.mp4">
          <SegmentTimeline>
            <S t="0" d="345600" r="2"/>
            <S t="1036800" d="316800"/>
          </SegmentTimeline>
        </SegmentTemplate>
```

```
    </Representation>
```

> **Note**
>
> If you use the `$Time$` format identifier in an output, you must also use it in every other output of the output group.

Specifying a minimum number of digits

For format identifiers that return a number, you can specify a minimum number of digits that the format identifier will resolve to. When you do, the service adds padding zeros before any value that would return fewer digits.

Use the following syntax to specify the number of digits: **`%0[number of digits]`**. Put this value just before the final $ of the format identifier.

For example, suppose that your video frame height is 720 and you want to specify a minimum of four digits, so that it appears in your file name as `0720`. To do that, use the following format identifier: **`$h%04$`**.

> **Note**
>
> Values that are too large to be expressed in the number of digits you specify resolve with more digits.

Working with output presets

Output presets speed up your job setup by providing groups of recommended transcoding settings. Job templates apply to an entire transcoding job; output presets apply to a single output of a transcoding job. For more information about job templates, see Templates.

You can use a system preset with settings specified for you, or you can create a custom preset with your own settings. You can create a custom preset from scratch, starting with only the default settings, or you can duplicate a system preset, adjust it to suit your workflow, and then save it as a custom preset.

Topics

Using output presets

When you specify the outputs of your MediaConvert job, you can use an output preset instead of choosing each output setting separately.

To specify a preset for an output using the MediaConvert console:

1. Create a job in the usual way, as described in Create a job.
2. Create output groups as described in Step 3: Create output groups.

> **Tip**
>
> Many jobs have one output for each type of device that will play the video created by the job.

3. On the **Create job** page, in the **Job** pane on the left, choose an output. Outputs are listed in the **Output groups** section, under their output group.

4. In the **Output settings** pane, choose an output preset from the **Preset** dropdown list. For more information about individual settings, choose the **Info** link next to each setting.

> **Note**
>
> The **Preset** dropdown list shows only the presets that work with the type of output group that the output is in.

5. For **Name modifier**, type a set of characters that will distinguish the files created from this output. For example, you might use `-DASH-lo-res` for the output in your DASH output group that has the lowest resolution.
6. Repeat these steps for each output in your job that you want to specify with a preset.
7. Finish creating the job as described in Create a job.

Listing and viewing output presets

You can list the system presets that are included with MediaConvert and the custom presets that you have added in an AWS Region. You can also view the settings for an individual preset.

To list output presets

1. Open the Output presets page in the MediaConvert console.
2. In the **Output presets** pane, from the **Presets** dropdown list, choose **Custom presets** or **System presets**.
3. If you are viewing system presets, you can optionally filter the list of presets by category. Do so by choosing from the **Category** dropdown list.
4. To display settings for an individual preset, choose the preset name from the list of presets.

Creating a custom preset

Output presets specify the settings that apply to a single output of a transcoding job. System presets have the output settings that are specified for you; custom presets have settings that are specified by you or by another user of your AWS account.

You can create a custom preset by individually specifying the settings, as described in this topic. Or you can create a custom preset by duplicating and modifying an existing preset, as described in [Creating a custom preset from a system preset](#).

To create a custom output preset

1. Open the Output presets page in the MediaConvert console.
2. In the **Output presets** pane, choose the **Create preset** button.
3. In the **Preset settings** pane, specify at a minimum the name of the new preset. Optionally, provide a description and a category.

   These values help you find the custom preset later. For more information see Listing and viewing output presets.
4. In the **Preset settings** pane, choose the container for the output.

   > **Tip**
   >
   > It's important to specify a container that is appropriate to the output type that you intend to create with the preset. When you choose a system or custom preset as part of creating a job, the console displays only the presets that specify a container that is valid for the output group.
5. Choose your output settings.

   For more information about each setting, choose the **Info** link located next to the setting or next to the heading for the group of settings.
6. Choose the **Create** button at the bottom of the page.

Creating a custom preset based on a system preset

MediaConvert doesn't allow you to modify system presets. If you want a preset that is like a system preset but slightly modified, you can duplicate the system preset, customize the settings, and save it as a custom preset.

To create a custom output preset based on a system preset

1. Open the Output presets page in the MediaConvert console.
2. In the **Output presets** pane, from the **Presets** dropdown list, choose **System presets**.

3. Choose the name of the system preset that is most like the custom preset that you want to create.
4. On the **Preset details** page, choose **Duplicate**.
5. On the **Create preset** page, specify a name for the new preset. Optionally, modify the description and category.

   These values help you find the custom preset later. For more information see Listing and viewing output presets.
6. Modify any output settings.

   For more information about each setting, choose the **Info** link that is located next to the setting or next to the heading for the group of settings.
7. Choose the **Create** button at the bottom of the page.

> **Note**
>
> This button looks like the **Create** button for creating a job, but in this context, choosing it creates the custom preset.

Modifying custom presets

You can adjust the settings and field values in your custom presets. You can't change system presets, but you can duplicate them and modify the duplicate, as described in Creating a custom preset from a system preset.

After you modify a preset, jobs that use the preset will run with the new settings, including the following:

- Jobs that directly specify the custom preset.
- Jobs that you create based on a template that uses the custom preset.
- Jobs that you duplicate from your job history that use the custom preset. The original job used the settings in the preset as they were at the time; the new job uses the current settings.

To modify a custom output preset

1. Open the Output presets page in the MediaConvert console.

2. Choose the name of the custom preset that you want to modify.
3. Adjust the settings.
4. Choose **Save**.

Deleting a custom preset

You can delete the custom presets that you added in an AWS Region. You can't delete the system presets.

To delete a custom preset

1. Open the Output presets page in the MediaConvert console.
2. Choose the name of the custom preset that you want to delete.
3. On the **Preset details** page, choose **Delete preset**.

Working with job templates

Job templates speed up your job setup by providing groups of recommended transcoding settings. Job templates apply to an entire transcoding job; output presets apply to a single output of a transcoding job. For more about output presets, see Presets.

You can use a system job template with settings specified for you, or you can create a custom job template with your own settings. You can create a custom job template from scratch, starting with only the default settings. Or you can duplicate a system job template, adjust it to suit your workflow, and then save it as a custom job template.

Topics

Using a job template to create a job

Job templates apply to an entire transcoding job and provide values for settings that stay the same across multiple jobs. You specify the input settings and the AWS Identity and Access Management (IAM) service role in the job itself. These values are not saved in the template because they are likely to vary from job to job.

To create a job using a job template

1. Open the Job templates page in the MediaConvert console.
2. In the **Job templates** pane, from the **Templates** dropdown list, choose **Custom job templates** or **System job templates**.

> **Note**
>
> Custom job templates appear only in the AWS Region where they are created. When you choose **Custom job templates**, you see only the job templates created in the AWS Region you chose at the beginning of this procedure.

3. Choose the name of the job template that you want to use.
4. On the **Job template details** page, choose **Create job**.
5. In the **Inputs** section of the **Job** pane, choose **Add**.
6. Specify your input video, audio, and captions settings.

> **Note**
>
> Make sure that you specify your audio and captions selectors in a way that corresponds to the outputs that are specified in the job template.

7. In the **Job settings** section of the **Job pane**, choose **Settings**.
8. In the **Job settings** pane, in the **IAM role** dropdown list, choose the service role that you created to grant permissions to MediaConvert to access your resources on your behalf. For instructions on creating this role, see Setting up IAM permissions .

Listing and viewing job templates

You can list the system job templates that are included with MediaConvert and the custom job templates that you have added in a AWS Region. You can also view the settings for an individual job template.

To list job templates

1. Open the Job templates page in the MediaConvert console.
2. In the **Job templates** pane, from the **Templates** dropdown list, choose **Custom job templates** or **System job templates**.
3. If you are viewing system job templates, you can optionally filter the list of job templates by category. Do so by choosing from the **Category** dropdown list.

4. To display settings for an individual job template, choose the job template name from the list of job templates.

Creating a custom job template

Job templates specify the settings that apply to all outputs of a transcoding job. System job templates have settings that are specified for you; custom job templates have settings that are specified by you or by another user of your AWS account.

You can create a job template by individually specifying the settings for each output. Or you can create a custom preset by specifying a preset for each output's settings, as described in Using output presets.

To create a custom job template

1. Open the Job templates page in the MediaConvert console.
2. In the **Job templates** pane, choose the **Create template** button.
3. In the **General information** pane, specify at a minimum the name of the new job template. Optionally, provide a description and a category.

   These values help you find the custom template later. For more information, see Listing and viewing job templates.
4. In the **Job template** pane, add inputs, output groups, outputs, and job-wide settings.

   The procedure for this is the same as described in Configuring jobs in MediaConvert, except that you don't specify the location and file name of your input, and you don't specify the IAM role that the service assumes so that it can access your resources.

   > **Note**
   >
   > If you set up outputs by referring to output presets, make sure to specify input audio and captions selectors to correspond with any output audio and captions that are specified in the preset. For example, if you use an output preset with three audio tracks that use audio selectors 1, 2, and 3, make sure that the input that you specify has audio selectors 1, 2, and 3.
5. Choose the **Create** button at the bottom of the page.

Modifying custom job templates

You can change the settings and field values in your custom job templates. You can't change system job templates, but you can modify the settings in jobs that you start from a system job template, before you create the job.

To modify a custom job template

1. Open the Job templates page in the MediaConvert console.
2. Choose the name of the custom job template that you want to modify.

   By default, the **Job templates** page is filtered to show only custom templates.
3. On the **Job template details** page, choose the **Update** button.
4. On the **Update job template** page, adjust the settings in the template, in the same way that you would when setting up a job.
5. Choose the **Update** button at the bottom of the **Update job template** page.

Deleting a custom job template

You can delete any custom job template. You can't delete system job templates.

To delete a job template

1. Open the Job templates page in the MediaConvert console.
2. Choose the name of the custom job template that you want to modify.

   By default, the **Job templates** page is filtered to show only custom templates.
3. On the **Job template details** page, choose the **Delete** button.

Working with queues in AWS Elemental MediaConvert

A queue is a set of resources that processes jobs.

To start a job, you submit it to a queue. AWS Elemental MediaConvert processes jobs that you submit to a queue in parallel. When all of your queue's resources are used, additional jobs stay in a `SUBMITTED` state until your queue's resources become available again. Use queues to manage the resources that are available to your account.

MediaConvert offers the following two queue types:

On-demand queues

With on-demand queues, you don't have to set up anything in advance. Your default queue is an on-demand queue. You pay rates that depend on the features that you use. For more information, see MediaConvert Pricing.

Reserved queues

With reserved queues, you pay for the transcoding capacity of the entire queue, regardless of how much or how little you use it. For more information, see MediaConvert Pricing.

All jobs run in a queue. If you don't specify a queue when you create your job, MediaConvert sends it to the default on-demand queue.

For information about how many queues you can create and how many jobs those queues can run, see Service quotas in the *AWS General Reference Guide.*

You can set up your job to move from one queue to another automatically if it stays in a `SUBMITTED` state for too long. For more information, see Queue hopping.

The time to complete a job varies significantly according to your input files and job settings. Accordingly, MediaConvert doesn't always complete jobs in the same order that you submit them.

How to specify a queue for your job

You specify a job's queue when you submit your job. The following tabs show different options for specifying a queue for your job.

Console

To specify a queue for your job by using the MediaConvert console, complete these steps:

1. Open the Create job page in the MediaConvert console.
2. Choose **Job management** from the **Job settings** menu.
3. Under **Queue**, choose a queue.

API, SDK, or the AWS CLI

To specify a job's queue by using the API, SDK, or the AWS Command Line Interface (AWS CLI), specify the `Queue` property. This property is a direct child of `Jobs`, which is in the top level of the JSON job specification. Set `Queue` to the ARN of your queue.

The following is an excerpt of a job settings JSON with an example queue specified.

```
{
 "Settings": {
  "OutputGroups": [...],
  "Inputs": [...]
 },
 "Queue": "arn:aws:mediaconvert:us-west-2:111122223333:queues/example"
}
```

For more information, see the MediaConvert API Reference.

The following topics provide information about how queues work in general, and about each type of queue specifically.

Topics

Using on-demand queues in AWS Elemental MediaConvert

Your default queue is an on-demand queue. On-demand queues differ from reserved queues in how AWS Elemental MediaConvert allocates transcoding resources for jobs and in how you pay. For more information, see MediaConvert Pricing. This section describes using multiple queues, creating additional queues, viewing queues, pausing or activating queues, and deleting queues.

Topics

Managing resources and testing performance

Manage resources to process even more jobs in parallel. Perform testing to optimize performance.

Resource allocation and job prioritization

By default, your account has one on-demand queue. You can create additional on-demand queues. The Service quotas documentation lists the maximum number of queues available to your account in an AWS Region.

Each queue can simultaneously process a number of jobs up to the maximum listed in the Service quotas documentation. For example, if you have five queues in us-east-1, you can process a total of 1000 jobs simultaneously. (In this example, you can process up to 200 jobs in any individual queue.)

When a job in an on-demand queue finishes, MediaConvert selects the next job to process based on the job's priority. You set the priority of a job when you create it. If more than one job has the highest priority, MediaConvert begins the one that you submitted first. For more information, see Job priority.

You can also organize your jobs with multiple on-demand queues. For example, you might run jobs for different workflows in separate queues. MediaConvert processes these jobs across multiple queues in parallel. You can use Tags to keep track of jobs with different workflows as well.

> **Note**
>
> If you need to process more jobs in parallel, instead of creating additional queues we recommend that you first request an increase to the number of jobs your queues can run. To do so, open the Service Quotas console, choose **Concurrent jobs per on-demand queue**, and select **Request quota increase**.

Performance testing

We recommend that you test any workflow with specific performance requirements. By default, MediaConvert optimizes the performance of your queue for the most common job types. If your workflow primarily includes a large number of jobs that complete quickly, or if you have a question about your queue's performance, contact AWS support.

How you pay for transcoding with on-demand queues

With on-demand queues, you pay based on usage. For pricing details, see AWS Elemental MediaConvert pricing.

Create an on-demand queue

AWS Elemental MediaConvert provides a default on-demand queue. A default queue is available in all AWS Regions. Other queues appear only in the AWS Region where they are created. To add additional resources to your account, you can create your own on-demand queues.

To learn how queues affect the way that MediaConvert allocates the processing of resources, see Managing resources and testing performance. The following tabs show different options for creating an on-demand queue.

Console

To create an on-demand queue by using the MediaConvert console:

1. Open the Queues page in the MediaConvert console.
2. Choose **Create queue**.
3. Enter a **Name**. Then optionally enter a **Description** and any **Tags**.
4. Choose **Create queue**.

AWS CLI

The following `create-queue` example creates a new on-demand queue.

```
aws mediaconvert create-queue \
 --region region-name-1 \
 --name Queue1 \
 --description "Example queue description." \
```

```
 --tags "KeyName1=string1,KeyName2=string2"
```

For more information about how to create an on-demand queue by using the AWS CLI, see the AWS CLI Command Reference.

Pause and reactivate on-demand queues

New queues default to an **Active** status and are available to process jobs immediately. You can optionally **Pause** a queue to stop processing any additional jobs. When you pause jobs, MediaConvert finishes processing jobs that are already running. If you submit a job to a paused queue, its status will remain in `SUBMITTED` until you change the queue's status back to **Active** or cancel the job.

The following tabs show how to change the status of an on-demand queue.

Console

To change the status of an on-demand queue by using the MediaConvert console:

1. Open the Queues page in the MediaConvert console.
2. In the **On-demand queues** section, select the queue.
3. Choose **Edit queue**.
4. Under **Status**, choose **Pause** or **Active**.
5. Choose **Save queue**.

AWS CLI

The following `update-queue` example pauses an active on-demand queue.

```
aws mediaconvert update-queue \
 --name Queue1 \
 --status PAUSED
```

The following `update-queue` example activates a paused on-demand queue.

```
aws mediaconvert update-queue \
 --name Queue1 \
 --status ACTIVE
```

For more information about how to change the status of an on-demand queue by using the AWS CLI, see the AWS CLI Command Reference.

Listing on-demand queues

You can list the queues that are associated with your AWS account and get details about those queues. The following tabs show different options for listing queues.

Console

To list your on-demand queues by using the MediaConvert console, open the Queues page. Select a queue to view its ARN.

AWS CLI

The following `list-queues` example lists all of your queues.

```
aws mediaconvert list-queues
```

The following JSON is an example list-queues response.

```
{
 "Queues": [
  {
   "Arn": "arn:aws:mediaconvert:us-west-2:111122223333:queues/Example",
   "CreatedAt": "2023-06-19T09:34:25-07:00",
   "LastUpdated": "2023-06-19T09:34:25-07:00",
   "Name": "Example",
   "PricingPlan": "ON_DEMAND",
   "ProgressingJobsCount": 0,
   "Status": "ACTIVE",
   "SubmittedJobsCount": 0,
   "Type": "CUSTOM"
  },
  {
   "Arn": "arn:aws:mediaconvert:us-west-2:111122223333:queues/Default",
   "CreatedAt": "2018-05-16T09:13:08-07:00",
   "LastUpdated": "2021-05-14T15:39:23-07:00",
   "Name": "Default",
   "PricingPlan": "ON_DEMAND",
   "ProgressingJobsCount": 0,
   "Status": "ACTIVE",
```

```
    "SubmittedJobsCount": 0,
    "Type": "SYSTEM"
   }
  ]
}
```

For more information about how to list queues by using the AWS CLI, see the AWS CLI Command Reference.

Delete an on-demand queue

You can delete any queue other than the default queue. You can't delete a queue that contains unprocessed jobs. The following tabs show how to delete an on-demand queue.

Console

To delete an on-demand queue by using the MediaConvert console:

1. Open the Queues page in the MediaConvert console.
2. Select the queue.
3. Choose **Delete queue**.

AWS CLI

The following `delete-queue` example deletes on-demand queue.

```
aws mediaconvert delete-queue \
 --name Queue1
```

For more information about how to delete an on-demand queue by using the AWS CLI, see the AWS CLI Command Reference.

Working with reserved queues in AWS Elemental MediaConvert

With reserved queues, you can purchase transcoding capacity for a 12-month period. The following topics provide information about working with reserved queues, such as creating and deleting queues, and allocating resources.

Topics

About reserved queues

Reserved queues differ from on-demand queues in how AWS Elemental MediaConvert allocates transcoding resources for jobs and in how you pay for your transcoding.

> **Note**
>
> There are a few features that you can't use with jobs that you send to a reserved queue. For more information, see Feature limitations with reserved queues.

Resource allocation and job prioritization with reserved queues

When you set up your reserved queue, you choose how many jobs it can run at once by specifying the number of reserved transcode slots (RTS) in the queue. For example, if you send five jobs to a reserved queue with two RTS, MediaConvert immediately begins processing the first two jobs that you submit, and it holds the other three in the queue. When one of the jobs that MediaConvert is processing finishes, the service begins processing the next job.

Each RTS has its own dedicated computing resources. Therefore, when MediaConvert processes a job that you send to a reserved queue, it takes the same amount of time to process whether the queue has one RTS or multiple RTS.

When a job in a reserved queue finishes, MediaConvert selects the next job to process based on the job's priority. You set the priority of a job when you create it. If more than one job has the highest priority, MediaConvert begins the one that you submitted first. For more information, see Job priority.

How you pay for transcoding with reserved queues

With reserved queues, you pay for the capacity in the queue regardless of whether you use it. When you set up a reserved queue, you make a 12-month commitment to a pricing plan. The pricing plan specifies a fixed number of reserved transcode slots (RTS). For more information, see MediaConvert Pricing.

> **Important**
>
> After you purchase your RTS, you can't cancel your 12-month commitment.

You can purchase additional capacity for a reserved queue that already has RTS. To purchase additional capacity, you extend your existing commitment with a new 12-month commitment for a larger number of RTS. The new commitment begins when you purchase the additional capacity. You can't decrease the number of RTS in your reserved queue.

When your pricing plan term expires, your reserved queue persists. You can still send jobs to it, but AWS Elemental MediaConvert doesn't run them.

About Auto Renew

You can set your pricing plan to auto renew. When your pricing plan term ends, AWS Elemental MediaConvert checks the auto renew status. If auto renew is enabled at that time, you automatically commit to another 12-month term for the same number of RTS at the same price. You can change the auto renew status at any time.

You can choose auto renew when you set up your queue. Anytime after that, you can change the auto renew status on the **Edit** page for the queue. For more information, see Create a reserved queue and Edit reserved queues.

About billing when jobs hop queues

When you set up queue hopping between a reserved queue and an on-demand queue, MediaConvert bills you according to the queue that it runs your job from. That is, if the job runs from your reserved queue, MediaConvert doesn't bill you for the job. If the job runs from your on-demand queue, MediaConvert bills you for the job at the on-demand rate.

Feature limitations with reserved queues

The following features are available only in jobs that you send to an *on-demand* queue. For jobs that you send to a *reserved* queue, you must disable the following features:

- 8k output resolution
- Automated ABR
- AV1 encoding
- Dolby Vision encoding
- FrameFormer frame rate conversion algorithm
- Accelerated transcoding

> **Note**
>
> Reserved queues cannot run accelerated jobs. However, you can submit a job to a reserved queue with **Accelerated transcoding** set as **Preferred**. When you do, if the job hops to an on-demand queue, it will run with acceleration. For more information, see Specify accelerated transcoding for hopped jobs.

Create a reserved queue

If you decide to purchase additional capacity for your reserved queue, you can. For more information, see Purchase additional capacity for a reserved queue. The following tabs show different options for creating a reserved queue.

Console

To create a reserved queue by using the MediaConvert console:

1. Open the Queues page in the MediaConvert console.
2. Choose **Create reserved queue**.
3. Enter a **Reserved queue name**. Optionally enter a **Description**.
4. Optionally, use the **Reserved transcode slots (RTS) calculator** to help determine how many RTS that you need.
5. In the **Commitment to purchase RTS for reserved queue** section, enter the number of reserved transcode slots (RTS) that you want to purchase.

6. Review and agree to the pricing and time commitment. **After you commit to your pricing plan, you can't cancel it.** Optionally, select **Auto renew yearly**.

7. Choose **Create reserved queue**. Then review your reserved queue details and choose **Purchase.**

AWS CLI

The following `create-queue` example creates a reserved queue with one RTS slot and a 12-month commitment.

```
aws mediaconvert create-queue \
 --region region-name-1 \
 --pricing-plan RESERVED \
 --reservation-plan-
settings "Commitment=ONE_YEAR,RenewalType=EXPIRE,ReservedSlots=1" \
 --name ReservedQueue1 \
 --description "Example reserved queue description." \
 --tags "KeyName1=string1,KeyName2=string2"
```

For more information about how to create a reserved queue by using the AWS CLI, see the AWS CLI Command Reference.

Purchase additional capacity for a reserved queue

To increase the number of jobs that your reserved queue can process at once, you can purchase additional capacity for it. To purchase additional capacity, you extend your existing commitment with a new 12-month commitment for a larger number of reserved transcode slots (RTS). The new commitment begins when you purchase the additional capacity. You can't decrease the number of RTS in your reserved queue. You can't cancel your commitment or revert to your original commitment after you increase the capacity.

The following tabs show how to purchase additional capacity for a reserved queue.

Console

To purchase additional capacity for a reserved queue by using the MediaConvert console:

1. Open the Queues page in the MediaConvert console.

2. In the **Reserved queues** section, select the reserved queue that you want to purchase additional capacity for.
3. Choose **Purchase additional capacity**.
4. In the **Commitment to purchase RTS for reserved queue** section, enter the number of reserved transcode slots (RTS) that you want to purchase. This number includes both the original amount of reserved transcode slots and the new additional capacity.
5. Review and agree to the pricing and time commitment. **After you commit to your pricing plan, you can't cancel or revert it.** Optionally, select **Auto renew yearly**.
6. Choose **Purchase additional capacity**. Then review your reserved queue details and choose **Purchase**.

AWS CLI

The following `update-queue` example adds an additional RTS to an existing reserved queue and begins a new 12-month commitment.

```
aws mediaconvert update-queue \
 --region region-name-1 \
 --reservation-plan-
settings "Commitment=ONE_YEAR,RenewalType=EXPIRE,ReservedSlots=2" \
 --name ReservedQueue1
```

For more information about how to update queues by using the AWS CLI, see the AWS CLI Command Reference.

Edit reserved queues

When you edit a queue, you can change the following:

- The description, which helps you identify it in the queue dashboard.
- The auto renew status of the queue's pricing plan for reserved transcode slots (RTS). For more information, see How you pay for transcoding with reserved queues.
- The paused or active status of the queue. Pausing the queue prevents the service from starting any more jobs until you reactivate the queue.

The following tabs show two options for editing a reserved queue.

Console

To edit a reserved queue by using the MediaConvert console:

1. Open the [Queues](#) page in the MediaConvert console.
2. In the **Reserved queues** section, select the reserved queue that you want to edit.
3. On the queue's page, choose **Edit queue**.
4. On the **Edit queue** page, make the changes that you want for the queue.
5. Choose **Save queue**.

AWS CLI

The following `update-queue` example updates the description and status of an existing reserved queue.

```
aws mediaconvert update-queue \
 --region region-name-1 \
 --description "Updated description." \
 --status "PAUSED" \
 --name ReservedQueue1
```

For more information about how to update queues by using the AWS CLI, see the AWS CLI Command Reference.

Listing reserved queues

You can list the AWS Elemental MediaConvert queues that are associated with your AWS account and get details about those queues. The following tabs show two options for listing your queues.

Console

To list your reserved queues by using the MediaConvert console, open the Queues page.

AWS CLI

The following `list-queues` example lists all of your queues.

```
aws mediaconvert list-queues
```

For more information about how list queues by using the AWS CLI, see the AWS CLI Command Reference.

Purchase transcoding capacity for an expired reserved queue

After your initial pricing plan term for reserved transcode slots (RTS) expires, your reserved queue persists without the capacity to run transcoding jobs. You can send jobs to the queue, but MediaConvert doesn't process them. To begin processing jobs through the queue again, you can set up a new pricing plan, which requires a new 12-month commitment.

The following tabs show two options for purchasing transcoding capacity for an expired reserved queue.

Console

To purchase transcoding capacity for an expired reserved queue by using the MediaConvert console:

1. Open the Queues page in the MediaConvert console.
2. Choose the reserved queue that you want to edit.
3. On the queue’s page, choose **Edit queue**.
4. On the **Edit queue** page, choose **Renew**.
5. In the **Commitment to purchase RTS for reserved queue** section, specify the number of reserved transcode slots (RTS) that you want to purchase.
6. Review and agree to the pricing and time commitment. **After you commit to your pricing plan, you can't cancel or revert it.** Optionally select **Auto renew yearly**.
7. Choose **Purchase additional capacity**. Then review your reserved queue details and choose **Purchase**.
8. Choose **Save queue**.

AWS CLI

The following `update-queue` example adds one RTS to an existing expired reserved queue and begins a new 12-month commitment.

```
aws mediaconvert update-queue \
```

```
 --region region-name-1 \
 --reservation-plan-
settings "Commitment=ONE_YEAR,RenewalType=EXPIRE,ReservedSlots=1" \
 --name ReservedQueue1
```

For more information about how to update queues by using the AWS CLI, see the AWS CLI Command Reference.

Delete a reserved queue

You can delete any queue other than the default queue. You can't delete a reserved queue that has an active pricing plan or that contains unprocessed jobs. The following tabs show two options for deleting a reserved queue.

Console

To delete a reserved queue by using the MediaConvert console:

1. Open the Queues page in the MediaConvert console.
2. Choose the reserved queue that you want to edit.
3. On the queue's page, choose **Delete queue**.

AWS CLI

The following `delete-queue` deletes an expired reserved queue.

```
aws mediaconvert delete-queue \
 --region region-name-1 \
 --name ReservedQueue1
```

For more information about how to update queues by using the AWS CLI, see the AWS CLI Command Reference.

Simulate a reserved queue

You can run a job in a *simulated* reserved queue to test its performance. When you do, MediaConvert runs your job from an on-demand queue with similar performance to what you will see with one RTS in a reserved queue. Take note of how long you job takes to complete and use this job completion time when calculating how many RTS you need.

Console

To simulate a job's reserved queue performance using the MediaConvert console:

1. Open the [Create job](#) page in the MediaConvert console.
2. Choose **Job management** on the left menu under **Job settings**.
3. Set **Simulate reserved queue** to **Enabled**.

API, SDK, or the AWS CLI

To simulate a job's reserved queue performance using the API, SDK, or AWS Command Line Interface (AWS CLI), set `SimulateReservedQueue` to `ENABLED`. This property is a direct child of `Jobs`, which is in the top level of the JSON job specification. The default value is `DISABLED`.

The following is an excerpt of a job settings JSON with `SimulateReservedQueue` set to `ENABLED`.

```
{
    "Settings": {
        "OutputGroups": [...],
        "Inputs": [...]
    },
  "SimulateReservedQueue": "ENABLED"
}
```

For more information, see the MediaConvert API Reference.

Job priority

Within a queue, AWS Elemental MediaConvert processes jobs in parallel until all the resources available to the queue are used. After a job finishes and the resources are again available in the queue, MediaConvert selects the next job to process based on the job's *priority*.

You specify priority when you create a job. MediaConvert processes jobs in each queue according to each job's priority, starting with the highest number. If more than one job has the highest value for priority, MediaConvert chooses among them by selecting the one that you submitted first.

MediaConvert doesn't stop the current job when you submit a job with a higher priority. When the running job is finished, MediaConvert starts the next job based on its relative priority in the queue.

After you create a job, you cannot change or update its priority. However, you can specify a new priority for jobs that hop queues. For additional information, see Set job priority for hopped jobs. The following tabs show different options for setting a job's priority.

Console

To set a job's priority in the MediaConvert console:

1. On the **Create job** page, choose **Job management**.
2. For **Priority**, enter a number from -50 to 50. MediaConvert processes jobs with the highest value for **Priority** first. If you don't specify a value, MediaConvert assigns the default value of 0.

API, SDK, or the AWS CLI

To set a job's priority in the API, SDK, or the AWS CLI, specify the `priority` property. This property is a direct child of `Jobs`, which is in the top level of the JSON job specification. Set the value of `Priority` to an integer in the range from -50 to 50, inclusive. The default value is 0.

The following is an excerpt of a job settings JSON with `Priority` set to 10.

```
{
    "Settings": {
        "OutputGroups": [...],
        "Inputs": [...]
    },
    "Priority": 10
}
```

For more information, see the MediaConvert API Reference.

Queue hopping

A job stays in a `SUBMITTED` state, waiting to be processed, until the queue that you submit it to has available resources. To prevent long wait times, you can configure your job to automatically move to another queue after a set amount of time. This is called *queue hopping*.

Keep the following definitions in mind with queue hopping.

Submission queue

The queue that you originally submit a job to is its *submission* queue.

Destination queue

The queue that your job moves to when it hops queues is its *destination* queue.

Wait time

The amount of time your job waits in its submission queue until it can hop to its destination queue.

Hop

A job *hops* when it moves from its submission queue to its destination queue after its wait time elapses. A job that moves queues is also referred to as a *hopped job*.

A common use case for queue hopping is to move jobs from a reserved queue to an on-demand queue during a spike in usage. For example, you might automatically move jobs that sit in a `SUBMITTED` state for longer than 10 minutes.

> **Note**
>
> When you set up queue hopping from a reserved queue to an on-demand queue, MediaConvert bills you according to the queue type that your job ultimately runs in. If your job runs in a reserved queue, MediaConvert doesn't bill you separately for the job because the cost is already covered by what you pay for your reserved queue. If your job runs in an on-demand queue, MediaConvert bills you for the job at the on-demand rate.

Topics

- Set up queue hopping
- View job history
- Set job priority for hopped jobs
- Specify accelerated transcoding for hopped jobs
- Queue hopping behavior with paused queues

Set up queue hopping

When you set up queue hopping, you specify the *submission queue*, the *wait time*, and the *destination queue*. Typically, the submission queue is a reserved queue and the destination queue is an on-demand queue. The following tabs show different options for setting up queue hopping.

Console

To set up queue hopping in the MediaConvert console:

1. On the **Create job** page, choose **Job management**.
2. Enable **Queue hopping**.
3. Enter the amount of time until your job can hop with **Wait minutes**.
4. For **Destination queue**, choose the queue that you want your job to hop to if it stays in the submission queue beyond its wait time.
5. Optionally, specify a new **Job priority** if your job hops to its destination queue. For more information, see Set job priority for hopped jobs.

API, SDK, or the AWS CLI

To set up queue hopping in the API, SDK, or the AWS CLI, configure the settings for queue hopping under `HopDestinations`. This property is a direct child of `Jobs`, which is in the top level of the JSON job specification.

The following is an excerpt of a job settings JSON that hops to an on-demand queue after 10 minutes.

```
{
    "Settings": {
        "OutputGroups": [...],
        "Inputs": [...]
    },
    "HopDestinations": [
        {
        "WaitMinutes": 10,
        "Queue": "arn:aws:mediaconvert:us-west-2:111122223333:queues/ondemandqueue",
        "Priority": 0
        }
    ]
```

```
}
```

For more information, see the MediaConvert API Reference.

View job history

When a job hops queues, the values for the settings `queue` and `priority` remain how you set them when you created the job. You can see the values for the job's post-hop destination and queue priority. The following tabs provide two options for viewing a job's history and queue priority.

Console

To see if your job hopped queues by using the MediaConvert console:

1. Open the Jobs page in the MediaConvert console.
2. Select a **Job ID**.
3. Find the **Queue transition** section that shows the job priority before and after hopping. That section also shows the epoch **Timestamp** for when the job hopped, the **Source queue**, and the **Destination queue**.

AWS CLI

The following `get-job` example returns a JSON response with information about your job.

```
aws mediaconvert get-job \
 --id 1234567890123-efg456
```

The following is an excerpt showing `QueueTransitions` in the JSON response when you run this command. The response shows your job's submission queue and destination queue.

```
"QueueTransitions": [
 {
  "Timestamp": 1672662636,
  "SourceQueue": arn:aws:mediaconvert:us-west-2:111122223333:queues/submissionqueue,
  "DestinationQueue": arn:aws:mediaconvert:us-west-2:111122223333:queues/
destinationqueue
 }
```

```
]
```

For more information about how to use the `get-job` command, see the [AWS CLI Command Reference](#).

Billing tags for hopped jobs

If you use billing tags on your jobs and set your billing tags source to **Queue**, the charges for your jobs are always listed under the tags for the submission queue. To track how much you were billed for a job that hops queues, you can set the billing tags source to **Job**. For more information about using tags to sort your AWS bill, see Setting up AWS Elemental MediaConvert resources for cost allocation through tagging.

> **Note**
>
> Cost allocation based on queue only applies to jobs that run in on-demand queues. When your submission queue is a reserved queue and your job hops to an on-demand queue, the charges for that on-demand job appear in your cost allocation report. If you don't put tags on your reserved queue, those charges appear in the report unsorted.

Listing hopped jobs

When you view your job, MediaConvert displays the queue that you submitted your job. For example, if you submit a job to `Queue1`, and it hops to `Queue2`, that job appears in lists that are filtered for `Queue1`. It doesn't appear in lists filtered for `Queue2`.

Set job priority for hopped jobs

When you set up a job for queue hopping, you can specify the priority for the job in the new queue. If you don't specify a new priority, the job keeps the priority number from its submission queue.

If you use different guidelines for choosing the values for `priority` between the two queues, make sure to specify a new priority value for the job in the destination queue.

For information about setting the job's priority within the submission queue, see Job priority.

The following tabs provide different options for setting the priority of a hopped job.

Console

To set the priority of a hopped job in the MediaConvert console:

1. Open the Create job page in the MediaConvert console.
2. Choose **Job management** from the **Job settings** menu.
3. Enable **Queue hopping**.
4. Enter the **Job priority** for when your job hops to its destination queue.
5. Enter both **Wait minutes** and **Destination queue**. For more information, see Queue hopping.

API, SDK, or the AWS CLI

You can set up a hopped jobs new priority by using the API, SDK, or the AWS CLI. To set it up, configure `Priority` under `HopDestinations`. This property is a direct child of `Jobs`, which is in the top level of the JSON job specification.

The following is an excerpt of a job settings JSON that sets a hopped job's priority to 25.

```
{
 "Settings": {
  "OutputGroups": [...],
  "Inputs": [...]
 },
 "HopDestinations": [
  {
   "WaitMinutes": 10,
   "Queue": "arn:aws:mediaconvert:us-west-2:111122223333:queues/ondemandqueue",
   "Priority": 25
  }
 ]
}
```

For more information, see the MediaConvert API Reference.

Specify accelerated transcoding for hopped jobs

To reduce the transcoding time for certain jobs, use accelerated transcoding. In most cases, you submit accelerated jobs to on-demand queues, because reserved queues cannot run accelerated jobs. However, you can submit a job with **Accelerated transcoding** set **Preferred** to a reserved queue. When you do, if the job hops to an on-demand queue, it will run with acceleration enabled. For more information about accelerated transcoding, see Accelerated transcoding in the MediaConvert User Guide.

The following tabs provide different options for setting accelerated transcoding.

Console

To set **Acceleration** to **Preferred** in MediaConvert console:

1. Open the Create job page in the MediaConvert console.
2. Choose **Job management** from the **Job settings** menu.
3. Under **Acceleration**, choose **Preferred** by using the dropdown list.

API, SDK, or the AWS CLI

To specify preferred acceleration by using the API, SDK, or the AWS CLI, configure `Mode` under `AccelerationSettings`. This property is a direct child of `Jobs`, which is in the top level of the JSON job specification.

The following is an excerpt of a job settings JSON that specifies queue hopping to an on-demand queue with accelerated transcoding.

```
{
 "Settings": {
  "OutputGroups": [...],
  "Inputs": [...]
 },
 "AccelerationSettings": {
  "Mode": "PREFERRED"
 },
 "HopDestinations": [
  {
   "WaitMinutes": 10,
   "Queue": "arn:aws:mediaconvert:us-west-2:111122223333:queues/ondemandqueue",
   "Priority": 25
  }
 ]
}
```

For more information, see the MediaConvert API Reference.

Queue hopping behavior with paused queues

Jobs don't hop from a queue while it's paused, but they hop freely to paused queues.

Hopping from a paused queue

Jobs don't hop from a queue while it's paused. Queue hopping behavior depends on how long the queue is paused. Consider these two situations:

You submit a job to a queue, pause the queue for longer than the queue hopping wait time, then reactivate it.

In this situation, whether the job hops depends on where the job is in the queue. If there are any jobs ahead of it in the queue, the job hops to the destination queue. If there are no jobs ahead of it in the queue, MediaConvert processes it without hopping.

For example, imagine that you submit a job to `Queue1` with a wait time of 15 minutes and a destination of `Queue2`. Five minutes after you submit the job, you pause `Queue1`. Ten minutes later, the job remains in `Queue1`. Half an hour after that, you activate `Queue1`. At that time, there are no jobs ahead of it in `Queue1`, so the job runs from `Queue1`.

You submit a job to a queue. You pause the queue and then reactivate it before the wait time passes.

In this situation, the time that the queue is paused doesn't affect queue hopping at all.

For example, imagine that you submit a job to `Queue1` with a wait time of 15 minutes and a destination of `Queue2`. Five minutes after you submit the job, you pause `Queue1`. One minute later, you reactivate `Queue1`. Nine minutes later (15 minutes after you submitted the job), there are still jobs ahead of it in the queue. Therefore, the job hops to `Queue2`, as though you hadn't paused the queue.

Hopping to a paused queue

Jobs hop freely from active queues to paused queues. For example, imagine that you submit a job to `Queue1` with a wait time of 15 minutes and a destination of `Queue2`. Then, five minutes after you submit the job, you pause `Queue2`. Ten minutes later (15 minutes after you submit the job), the job hops to `Queue2` and remains there, waiting until you activate the queue.

MediaConvert features

This chapter contains detailed information for implementing AWS Elemental MediaConvert features. You set up these features when you create a job.

Topics

Color conversion with 3D LUTs

When you convert a video from one color space to another, AWS Elemental MediaConvert automatically maps colors from your input color space to your output color space. To optionally specify your own custom color mapping, use **3D LUTs** (3D lookup tables).

3D LUTs contain color mapping information for a specific input or set of inputs. You receive 3D LUTs as .cube files from your color grader as part of your video production workflow.

3D LUTs job setting requirements

When you include 3D LUTs as part of your MediaConvert job, you must also include the following settings:

Input color space

Specify which inputs use this 3D LUT, according to the input's color space.

Input mastering luminance

(Optional) Include **Input mastering luminance** only when your input has an **HDR10** or **P3D65 (HDR)** color space. Otherwise, keep blank. Use to select between inputs with different mastering luminances.

Output color space

Specify which outputs use this 3D LUT, according to the output's color space.

Output mastering luminance

(Optional) Include **Output mastering luminance** only when your output has an **HDR10** or **P3D65 (HDR)** color space. Otherwise, keep blank. Use to select between outputs with different mastering luminances.

.cube file

Specify an Amazon S3, HTTP, or HTTPS URL for your .cube file. MediaConvert accepts .cube files up to 8MB in size.

Color corrector

Specify an output color space in the **Color corrector** preprocessor for your video output.

Using 3D LUTs

MediaConvert only applies color space conversions with your 3D LUTs to sets of inputs and outputs that match the settings that you specify. You can apply custom color mapping to some of your outputs, and apply default color mapping to others.

For example, you might include a 3D LUT that specifies a **Rec. 601** input color space and a **Rec. 709** output color space. When your job settings meet the requirements listed in the previous section, MediaConvert applies your custom color mapping for all **Rec. 601** inputs that result in **Rec. 709** outputs.

Specify a separate 3D LUT for each combination of input and output color space conversion that you need. In a single job, you can include up to 8 different 3D LUT settings. MediaConvert uses default color mapping for inputs or outputs with color spaces not included in your 3D LUT or job settings.

MediaConvert console

To specify a 3D LUT by using the MediaConvert console:

1. Open the [Create job](#) page in the MediaConvert console.
2. On the **Create job** page, provide transcode instructions and job settings. For more information, see [Configuring jobs in MediaConvert](#).
3. Turn on the **Color corrector** preprocessor under the **Encoding settings** of your video output.
4. Specify an output color space under **Color space conversion**.
5. Choose **Global processing** under **Job settings**.
6. Turn on **3D LUTs**.
7. **Under Input color space**, enter the color space for your input or inputs that you want to apply this 3D LUT to.
8. (Optional) If you entered **HDR10** or **P3D65 (HDR)** for **Input color space**, optionally enter a value for **Input mastering luminance**. Use to select inputs with a specific luminance. Enter the same value that you enter for **Maximum luminance**, which is under the input's **Video selector**, **Video correction**, settings.

   If you enter `0` for **Input mastering luminance**, or keep blank, your 3D LUT will apply to all of your **HDR10** or **P3D65 (HDR)** inputs.

9. **Under Output color space**, enter the color space for your output or outputs that you want to apply this 3D LUT to.
10. (Optional) If you entered **HDR10** or **P3D65 (HDR)** for **Output color space**, optionally enter a value for **Output mastering luminance**. Use to select outputs with a specific luminance. Enter the same value that you enter for **Maximum luminance**, which is under the output's **Video** settings.

    If you enter `0` for **Output mastering luminance**, or keep blank, your 3D LUT will apply to all of your **HDR10** or **P3D65 (HDR)** outputs.
11. Under **3D LUT file input**, enter the URL for your 3D LUT .cube file.

API, SDK, or AWS Command Line Interface (AWS CLI)

To specify a 3D LUT by using the API, SDK, or AWS Command Line Interface (AWS CLI), include `ColorConversion3DLUTSettings` in your job settings JSON.

The following is an excerpt of a job settings JSON that specifies a 3D LUT for an **HDR 10** to **P3D65 (HDR)** workflow:

```
{
  "Settings": {
    "Inputs": [...],
    "OutputGroups": [
      {
        "Name": "File Group",
        "OutputGroupSettings": {...},
        "Outputs": [
          {
            "VideoDescription": {
              "CodecSettings": {... },
              "VideoPreprocessors": {
                "ColorCorrector": {
                  "ColorSpaceConversion": "FORCE_P3D65_HDR",
                  "MaxLuminance": 3000
                }
              }
            },
            "AudioDescriptions": [...],
            "ContainerSettings": {...}
          }
        ]
```

```
      }
    ],
    "ColorConversion3DLUTSettings": [
      {
        "InputColorSpace":"HDR10",
        "InputMasteringLuminance": 0,
        "OutputColorSpace": "P3D65HDR",
        "OutputMasteringLuminance": 3000,
        "FileInput": "s3://DOC-EXAMPLE-BUCKET/HDR10_to_P3D65HDR.cube"
      }
    ]
  }
}
```

Using accelerated transcoding in AWS Elemental MediaConvert

AWS Elemental MediaConvert jobs that create premium content may have high computational requirements and can take longer to complete. Such jobs may include Ultra High Definition (UHD) or High Dynamic Range (HDR) content. To reduce the transcoding time required to run these jobs, you can use accelerated transcoding. Consider using accelerated transcoding for jobs that would otherwise take 10 minutes or longer to run.

For example, jobs that generate the following assets might benefit from accelerated transcoding:

- Ultra High Definition content
- High dynamic range content in HEVC
- Any long-duration, visually complex video

> **Note**
>
> Accelerated transcoding is a Professional tier feature. You pay more per minute of transcoded output for outputs that use Professional tier features. For more information about MediaConvert pricing tiers, see MediaConvert pricing.

Topics

- Setting up accelerated transcoding in AWS Elemental MediaConvert
- Job limitations for accelerated transcoding in AWS Elemental MediaConvert

- Example accelerated transcoding JSON job for AWS Elemental MediaConvert

Setting up accelerated transcoding in AWS Elemental MediaConvert

You set up accelerated transcoding for your AWS Elemental MediaConvert jobs in the same way that you set up unaccelerated jobs, except that you enable acceleration.

> **Note**
>
> We recommend that you use a dedicated transcoding queue for your accelerated transcoding jobs. This will provide isolation between the resources that you use for your accelerated jobs and your other jobs.

To set up your transcoding job with accelerated transcoding (console)

1. Set up your transcoding job as usual. For more information, see the section called "Configuring jobs".

   Make sure that your job input files and output settings conform to the limitations and requirements listed in Job limitations for accelerated transcoding in AWS Elemental MediaConvert.
2. Change your timecode settings from the default value **Embedded** to **Start at zero**.
   a. On the **Create job** page, in the **Job** pane on the left, under **Job settings**, choose **Settings**.
   b. In the **Timecode configuration** pane, for **Source**, choose **Start at 0**.
   c. On the **Create job** page, in the **Job** pane on the left, under **Inputs**, choose the input.
   d. In the **Video selector** pane, for **Timecode source**, choose **Start at 0**.
3. If you don't already have a dedicated queue for accelerated transcoding jobs, create one. For more information, see Create an on-demand queue.
4. On the **Create job** page, in the **Job** pane on the left, in the **Job Settings** section, choose **Settings**.
5. For **Acceleration**, choose **Enabled** or **Preferred**.

   With both **Enabled** and **Preferred**, if your input files and transcoding settings are compatible with accelerated transcoding, MediaConvert runs the job with accelerated transcoding.

If your input files or transcoding settings aren't compatible with accelerated transcoding, MediaConvert handles the job differently, depending on the value that you set for **Acceleration**:

- **Enabled** – The service fails the incompatible job.
- **Preferred** – The service runs the job without accelerated transcoding.

  Setting **Acceleration** to **Preferred** incurs Professional tier pricing only when MediaConvert runs the job with accelerated transcoding.

For more information about what files and settings are compatible with accelerated transcoding, see Job limitations for accelerated transcoding in AWS Elemental MediaConvert.

If you use the API or an SDK, you can find this setting in the JSON file of your job. The setting name is AccelerationMode, under AccelerationSettings.

Job limitations for accelerated transcoding in AWS Elemental MediaConvert

Before you enable accelerated transcoding, make sure that your job conforms to the following requirements and limitations.

Video inputs supported with accelerated transcoding

The following table shows the video input codecs and containers that MediaConvert supports with accelerated transcoding.

Container	Video Codecs Supported with Container
HLS (MPEG-2 TS segments)	AVC (H.264), HEVC (H.265)
IMF	JPEG 2000 (J2K)
Matroska	AVC (H.264), MPEG-2, MPEG-4 part 2, VC-1
MPEG Transport Streams	AVC (H.264), HEVC (H.265), MPEG-2, VC-1

Container	Video Codecs Supported with Container
MPEG-4	AVC Intra 50/100, AVC (H.264), HEVC (H.265), MPEG-2
MXF	Apple ProRes, AVC Intra 50/100, AVC (H.264), JPEG 2000 (J2K), MPEG-2, SonyXDCam, SonyXDCam (as an MPEG-2 variant only), VC-3
QuickTime	Apple ProRes, AVC Intra 50/100, AVC (H.264), JPEG 2000 (J2K), MPEG-2

> **Note**
>
> For HLS inputs, your package must conform to the requirements listed in HLS input package requirements.

Video outputs supported with accelerated transcoding

The following table shows the video output codecs and containers that MediaConvert supports with accelerated transcoding.

Container	Codecs Supported with Container
CMAF	AV1, AVC (H.264), HEVC (H.265)
DASH	AV1, AVC (H.264), HEVC (H.265)
HLS	AVC (H.264), HEVC (H.265)
MPEG-2 TS	AVC (H.264), HEVC (H.265)
MPEG-4 (.mp4)	AV1, AVC (H.264), HEVC (H.265)
MPEG-4 Flash (.f4v)	AVC (H.264)
MXF	AVC (H.264)

Container	Codecs Supported with Container
QuickTime	AVC (H.264)
Smooth (ISMV)	AVC (H.264)
WebM	VP8, VP9
Raw (no container)	AVC (H.264), HEVC (H.265)

> **Note**
>
> For MPEG-2 TS outputs, to use accelerated transcoding, you must change the default value of **CBR** for **Transport stream settings** > **Rate mode** to **VBR**.

Output restrictions

- Your output must have a video component. MediaConvert doesn't support accelerated transcoding with audio-only outputs.
- Use only supported combinations of container and codec. The preceding table shows the supported video output codecs and containers.
- Use any output audio codec other than Dolby Atmos. MediaConvert doesn't support accelerated transcoding with Dolby Atmos.
- Use only supported output captions formats. For accelerated transcoding, MediaConvert doesn’t support the following output captions formats:
  - Not supported: Burn-in
  - Not supported: SCTE-20

  For a list of supported captions formats, see [Supported captions workflows, input captions in the same file as video](#).

Transcoding features not supported with accelerated transcoding

- Avail blanking
- Motion image inserter

- Interpolated frame rate conversion
- VBI passthrough
- Timecode passthrough
- SEI timecode
- Timecode anchor
- Telecine output
- Inverse telecine output
- Embedded timecode source

> **Note**
>
> With accelerated transcoding, you can set the input setting **Timecode source** to **Embedded**, but not the job-wide setting **Source** under **Timecode configuration**.

- Black video generator
- Pad video
- Inputs with variable frame rates
- Inputs with discontinuities

Example accelerated transcoding JSON job for AWS Elemental MediaConvert

The following example JSON job converts an MP4 file to another MP4 file that is 720p encoded with H.265. To use this example, replace the settings with your values. At minimum, you must provide values for the following settings:

- **`Role`**: This is the AWS Identity and Access Management (IAM) role that you set up to give AWS Elemental MediaConvert permission to access your input and output Amazon S3 buckets and to access Amazon API Gateway on your behalf. For information about setting up this role, see Set up IAM permissions in the *AWS Elemental MediaConvert User Guide*.
- **`Destination`**: The Amazon S3 bucket where you want MediaConvert to store your output file. Make sure to include a trailing backslash, as in the example.

- **`InputClippings`**: Define the clips that you want transcoded by specifying sets of values for `StartTimecode` and `EndTimecode`. Alternatively, you can remove `InputClippings` entirely to transcode the entire asset.
- **`FileInput`**: Specify the file name and location for your input file. Your file input can be an Amazon S3 object or an HTTP URL.

  Accelerated transcoding is supported with jobs that have only a single input; you can't do input stitching.

```
{
  "Role": "arn:aws:iam::123456789012:role/MediaConvert_Role",
  "AccelerationSettings" : {
    "Mode" : "ENABLED"
  },
  "UserMetadata": {
      "job" : "Acceleration"
  },
  "Settings": {
    "TimecodeConfig": {
      "Source": "ZEROBASED"
    },
    "OutputGroups": [
      {
        "Name": "File Group",
        "Outputs": [
          {
            "ContainerSettings": {
              "Container": "MP4",
              "Mp4Settings": {
                "CslgAtom": "EXCLUDE",
                "FreeSpaceBox": "EXCLUDE",
                "MoovPlacement": "NORMAL"
              }
            },
            "VideoDescription": {
              "Width": 1280,
              "ScalingBehavior": "DEFAULT",
              "Height": 720,
              "VideoPreprocessors": {
                "TimecodeBurnin": {
                  "FontSize": 32,
```

```
            "Position": "TOP_CENTER"
          }
        },
        "TimecodeInsertion": "DISABLED",
        "AntiAlias": "ENABLED",
        "Sharpness": 50,
        "CodecSettings": {
          "Codec": "H_265",
          "H265Settings": {
            "InterlaceMode": "PROGRESSIVE",
            "ParNumerator": 1,
            "NumberReferenceFrames": 3,
            "FramerateDenominator": 1001,
            "GopClosedCadence": 1,
            "AlternateTransferFunctionSei": "DISABLED",
            "HrdBufferInitialFillPercentage": 90,
            "GopSize": 48,
            "Slices": 4,
            "GopBReference": "ENABLED",
            "HrdBufferSize": 20000000,
            "SlowPal": "DISABLED",
            "ParDenominator": 1,
            "SpatialAdaptiveQuantization": "ENABLED",
            "TemporalAdaptiveQuantization": "ENABLED",
            "FlickerAdaptiveQuantization": "DISABLED",
            "Bitrate": 10000000,
            "FramerateControl": "INITIALIZE_FROM_SOURCE",
            "RateControlMode": "CBR",
            "CodecProfile": "MAIN_MAIN",
            "Tiles": "ENABLED",
            "Telecine": "NONE",
            "FramerateNumerator": 24000,
            "MinIInterval": 0,
            "AdaptiveQuantization": "HIGH",
            "CodecLevel": "LEVEL_5",
            "SceneChangeDetect": "ENABLED",
            "QualityTuningLevel": "SINGLE_PASS_HQ",
            "FramerateConversionAlgorithm": "DUPLICATE_DROP",
            "UnregisteredSeiTimecode": "DISABLED",
            "GopSizeUnits": "FRAMES",
            "ParControl": "SPECIFIED",
            "NumberBFramesBetweenReferenceFrames": 3,
            "TemporalIds": "DISABLED",
            "SampleAdaptiveOffsetFilterMode": "ADAPTIVE"
```

```
            }
          },
          "AfdSignaling": "NONE",
          "DropFrameTimecode": "ENABLED",
          "RespondToAfd": "NONE",
          "ColorMetadata": "INSERT"
        },
        "AudioDescriptions": [
          {
            "AudioTypeControl": "FOLLOW_INPUT",
            "CodecSettings": {
              "Codec": "AAC",
              "AacSettings": {
                "AudioDescriptionBroadcasterMix": "NORMAL",
                "Bitrate": 160000,
                "RateControlMode": "CBR",
                "CodecProfile": "LC",
                "CodingMode": "CODING_MODE_2_0",
                "RawFormat": "NONE",
                "SampleRate": 48000,
                "Specification": "MPEG4"
              }
            },
            "LanguageCodeControl": "FOLLOW_INPUT",
            "AudioType": 0
          }
        ],
        "Extension": "mp4",
        "NameModifier": "1280x720"
      }
    ],
    "OutputGroupSettings": {
      "Type": "FILE_GROUP_SETTINGS",
      "FileGroupSettings": {
        "Destination": "s3://DOC-EXAMPLE-BUCKET1/accelerated/"
      }
    }
  }
],
"AdAvailOffset": 0,
"Inputs": [
  {
    "InputClippings": [
      {
```

```
            "EndTimecode": "01:00:00:00",
            "StartTimecode": "00:00:00:00"
          }
        ],
        "AudioSelectors": {
          "Audio Selector 1": {
            "Offset": 0,
            "DefaultSelection": "DEFAULT",
            "ProgramSelection": 1
          }
        },
        "VideoSelector": {
          "ColorSpace": "FOLLOW"
        },
        "FilterEnable": "AUTO",
        "PsiControl": "USE_PSI",
        "FilterStrength": 0,
        "DeblockFilter": "DISABLED",
        "DenoiseFilter": "DISABLED",
        "TimecodeSource": "ZEROBASED",
        "FileInput": "s3://DOC-EXAMPLE-BUCKET/SampleVideo_h264_StereoAudio.mp4"
      }
    ]
  }
}
```

Audio descriptions

AWS Elemental MediaConvert supports two different workflows for including broadcast audio descriptions in your output.

You can mix an audio description with other audio content if your input contains an *audio description audio signal* and an *audio description data stream*. An audio description audio signal is a spoken description of a video, made for people who cannot see the visual content. An audio description data stream contains fade and pan data used by an encoder. MediaConvert uses this data stream to temporarily lower the volume of other audio channels while an audio description is active.

If your input already has pre-mixed audio descriptions instead of an audio signal and data stream channel, you can signal it to downstream systems by writing audio description metadata in your output.

For more information about audio descriptions, see BBC WHP 198 and BBC WHP 051.

Topics

Requirements for mixing audio descriptions

When you mix audio descriptions, jobs have the following requirements:

Input

You need an input with an audio description audio signal on one audio channel and an audio description data stream on another.

Audio remixing

To mix audio descriptions across multiple audio selectors, for example if you have a sidecar audio file, enable **Manual audio remixing** in your output.

Otherwise, to apply audio description mixing to a single input audio selector, you can enable **Input remix controls** in your input instead. Keep in mind that this mixing will not apply across multiple audio selectors or across audio selector groups.

Mixing audio descriptions

The following sections describe how to configure your job settings to mix audio descriptions.

MediaConvert console

To mix audio descriptions in your output by using the MediaConvert console:

1. Open the Create job page in the MediaConvert console.
2. Add an input that has an audio description audio signal and an audio description data stream.
3. Add an output with at least one audio track.
4. In the output audio track, expand **Advanced** and enable **Manual audio remixing**.

5. Enter the channel that has your audio description audio signal in **Audio description audio channel**.
6. Enter the channel that has your audio description data stream in **Audio description data channel**.
7. Specify the total number of **Input channels**. For example, if your input has four input channels including left, right, audio description audio signal, and audio description data stream, select **4** .
8. Specify the total number of **Output channels**. For example, for a stereo output select **2**.
9. Specify mixing levels under **Channel mapping**. For example, to mix an audio description audio signal from channel 3 across left and right channels in a stereo output, and to mute the data stream, enter the following:

   Channel mapping

Inputs	Outputs	
	L (0)	R (1)
1	**0**	**-60**
2	**-60**	**0**
3	**0**	**0**
4	**-60**	**-60**

> **Note**
>
> When played over speakers, audio description data streams sound like noise. Mute the data stream in your output by setting its channel mapping to **-60**, as shown in the previous channel mapping example table.

API, SDK, or AWS Command Line Interface (AWS CLI)

The following is an excerpt of a job settings JSON that specifies audio description mixing for a stereo output. Note that the audio description audio signal is in input channel 3, and the audio description data stream is in input channel 4:

```
{
  "Settings": {
    "Inputs": [],
    "OutputGroups": [
      {
        "Name": "File Group",
        "OutputGroupSettings": {
          "Type": "FILE_GROUP_SETTINGS",
          "FileGroupSettings": {}
        },
        "Outputs": [
          {
            "VideoDescription": {},
            "AudioDescriptions": [
              {
                "CodecSettings": {
                  "Codec": "AAC",
                  "AacSettings": {
                    "Bitrate": 96000,
                    "CodingMode": "CODING_MODE_2_0",
                    "SampleRate": 48000
                  }
                },
                "AudioSourceName": "Audio Selector 1",
                "RemixSettings": {
                  "ChannelMapping": {
                    "OutputChannels": [
                      {
                        "InputChannelsFineTune": [
                          0,
                          -60,
                          0,
                          -60
                        ]
                      },
                      {
                        "InputChannelsFineTune": [
                          -60,
                          0,
                          0,
                          -60
                        ]
                      }
```

```
                    ]
                  },
                  "ChannelsIn": 4,
                  "ChannelsOut": 2,
                  "AudioDescriptionAudioChannel": 3,
                  "AudioDescriptionDataChannel": 4
                }
              }
            ],
            "ContainerSettings": {
              "Container": "MP4",
              "Mp4Settings": {}
            }
          }
        ]
      }
    ]
  }
}
```

Requirements for pre-mixed audio descriptions

If you have an input that already has pre-mixed audio descriptions, and does not contain an *audio description audio signal* or an *audio description data stream*, set **Audio description broadcaster mix** to **Broadcaster mixed AD**.

When you do, MediaConvert writes metadata in your output that signals to downstream systems that it contains broadcaster mixed audio descriptions.

Specifying pre-mixed audio descriptions

The following sections describe how to configure your job settings to write metadata in your output signalling it contains pre-mixed audio descriptions.

MediaConvert console

To write broadcaster mixed audio description metadata in your output by using the MediaConvert console:

1. Open the [Create job](https://console.aws.amazon.com/mediaconvert/home#/jobs/create) page in the MediaConvert console.
2. Add an input that has pre-mixed audio description.
3. Add an output with at least one audio track.

4. In the output audio track, set **Audio description broadcaster mix** to **Broadcaster mixed AD**.
5. (Optional) If you enable Manual audio remixing, keep **Audio description audio channel** and **Audio description data channel** blank, as these channels will not be present in your input.

API, SDK, or AWS Command Line Interface (AWS CLI)

The following is an excerpt of a job settings JSON that specifies pre-mixed audio descriptions for a stereo output:

```
{
  "Settings": {
    "Inputs": [],
    "OutputGroups": [
      {
        "Name": "File Group",
        "OutputGroupSettings": {
          "Type": "FILE_GROUP_SETTINGS",
          "FileGroupSettings": {}
        },
        "Outputs": [
          {
            "VideoDescription": {},
            "AudioDescriptions": [
              {
                "CodecSettings": {
                  "Codec": "AAC",
                  "AacSettings": {
                    "Bitrate": 96000,
                    "CodingMode": "CODING_MODE_2_0",
                    "SampleRate": 48000,
                    "AudioDescriptionBroadcasterMix": "BROADCASTER_MIXED_AD"
                  }
                }
              }
            ],
            "ContainerSettings": {
              "Container": "MP4",
              "Mp4Settings": {}
            }
          }
        ]
      }
```

```
    ]
  }
}
```

Using automated ABR in MediaConvert

With automated ABR, AWS Elemental MediaConvert sets up your adaptive bitrate (ABR) stack for you. MediaConvert chooses the right number of renditions and the resolution for each, based on the input video. MediaConvert minimizes the total minutes of transcoded output by eliminating renditions that increase bitrate without providing increased video quality. Automated ABR also maximizes video quality at various bitrates by employing the quality-defined variable bitrate (QVBR) rate control mode.

Topics

How automated ABR works

With all adaptive bitrate (ABR) streaming, the end viewer's player device adjusts which rendition of the package it downloads based on the available bandwidth. For example, a viewer with access to high-quality wifi would automatically see a high-bitrate rendition. When they move to a location with only limited bandwidth, their player device automatically switches over to a lower-bitrate rendition. How well the adaptive streaming works depends on the construction of the ABR stack and how well that construction suits the content of the specific video. For example, with a fast-moving, visually complex asset, your ABR stack might include two 720p outputs with different bitrates. If you used that same stack setup with a simple cartoon, those two outputs would likely look the same. The extra encoding, storage, and distribution would cost money without conferring any benefit to the end viewer.

When you run an automated ABR job, MediaConvert maximizes the video quality that the end viewer sees based on their available bandwidth. It does this by analyzing a broad set of possible renditions and eliminating any that increase required bandwidth without increasing video quality.

When you run your job, MediaConvert analyzes the content of your input video and chooses the number of renditions and the characteristics of each rendition for you.

You can use the MediaConvert console to run your job without setting anything. There are three categories of optional settings you can specify if you choose to:

- Limits on your adaptive bitrate (ABR) stack.
- Limits that apply to the renditions in the ABR stack. These output-level limits apply to all renditions in the stack.
- All other encoding settings.

  For these settings, MediaConvert uses default values unless you specify something different. This works the same as with outputs that don't use automated ABR, except that whatever values you set apply to all renditions in the stack. For example, if you set **Profile** to **High 10-bit**, every rendition will have that codec profile.

Settings that apply to the ABR stack

You can set the following limits that relate to the whole ABR stack:

- **Max renditions**: This is the upper limit for the number of renditions in your ABR stack. The number of renditions in your stack might be less than this, but won't be more.

  You can specify a number from 3–15. If you don't specify this, the default maximum is 15.
- **Max ABR bitrate**: The maximum average bitrate for the highest-bitrate rendition in your stack.

  This is the rendition that is delivered to viewers with the fastest internet connections. Use to limit the total bytes that are egressed to viewers who receive the highest-bitrate rendition.

  If you don't specify this, the default maximum is 8 mb/s.

  The average bitrate of your highest-quality rendition will be equal to or below this value, depending on the quality, complexity, and resolution of your content. The instantaneous maximum bitrate can vary above the value that you specify.
- **Min ABR bitrate**: The minimum average bitrate for the lowest-quality rendition in your stack.

  This is the rendition that is delivered to viewers with the slowest internet connections.

  If you don't specify this, the default minimum is 600 kb/s.

The instantaneous minimum bitrate can vary below the value that you specify.

Settings that apply to renditions in the stack

You can set the following limits that relate to properties of the renditions in the stack:

- **Max resolution**: This is the maximum resolution of your highest-bitrate rendition. When you set this value, choose the resolution of the highest quality device that you expect end viewers to use. MediaConvert won't create a rendition with a resolution larger than this.

  If you don't specify this, the default maximum is the resolution of your input video.

  If you specify a value larger than your input video's resolution, MediaConvert uses your input resolution as the maximum. MediaConvert won't use a resolution larger than the input for any rendition because upscaling the input resolution would add bandwidth without adding video quality.
- **Max frame rate**: MediaConvert uses this value as the frame rate for the for highest-bandwidth rendition in your stack. Depending on the input, this might be the frame rate for all renditions. When your input frame rate is high, MediaConvert might halve the frame rate for lower-bandwidth renditions. For example, if your input frame rate is 60 fps, MediaConvert might use 30 fps for some of the lower-bandwidth renditions, and perhaps 15 fps for the very lowest.

  If you don't specify this value, the default maximum is your input frame rate.

For these settings, MediaConvert determines these values for each rendition automatically:

- **Quality tuning level**: MediaConvert encodes all renditions with **Multi pass HQ**.

  This behavior is automatic in the MediaConvert console but not when you submit your job programmatically. When you set up your JSON job specification without using the MediaConvert console, you must explicitly set `qualityTuningLevel` to `MULTI_PASS_HQ`.
- **Rate control mode**: MediaConvert encodes all renditions with [QVBR](#) rate control mode.

  This behavior is automatic in the MediaConvert console but not when you submit your job programmatically. When you set up your JSON job specification without using the MediaConvert console, you must explicitly set `rateControlMode` to `QVBR`.
- These QVBR required settings:

  - **QVBR quality level**:
  - **Max bitrate**
  - **Max average bitrate**
- **HDR buffer size**
- **HDR buffer initial fill**

Creating an automated ABR stack

When you know whether you want to specify values for the optional limits, use one of the following procedures to set up automated ABR in your job. For more information about these optional settings, see How automated ABR works.

Console

To set up an automated ABR job using the MediaConvert console:

1. Begin setting up your job as usual. For more information, see Getting started with MediaConvert .

   Instead of following the general instructions for setting up output groups and outputs, follow this procedure.
2. Enable accelerated transcoding. This is optional, but we strongly recommend it. For more information, see Accelerated transcoding.
3. On the **Create job** page, in the **Job** pane on the left, next to **Output groups**, choose **Add**.
4. Choose a streaming output group type: Apple HLS, DASH ISO, Microsoft Smooth Streaming, or CMAF.
5. Below the **Group settings** section, enable **Automated ABR**.
6. Optionally, specify any Automated ABR rules.
7. For the settings in the **Automated ABR** section, optionally specify values. These are the settings that specify limits that relate to the whole ABR stack. If you choose to keep the default values, you should get good results.

   For more information, see How automated ABR works.
8. From the **Job** pane on the left, below your output group, choose **Automated ABR base output**.

9. On the right, in the **Base encoding settings for auto-generated ABR renditions** section, optionally specify values for the limits that relate to properties of the renditions in the stack. If you choose to keep the default values, you should get good results.

   For more information, see How automated ABR works.

10. Optionally, specify values for encoding settings that aren't directly related to automated ABR, such as codec profile and level. Values you specify apply to all renditions in the stack. If you choose to keep the default values, you should get good results.

    For information about each individual setting, choose the **Info** link next to the setting in the MediaConvert console to view the setting description.

11. If your workflow requires video and audio in separate unmuxed outputs, remove **Audio 1** from **Base encoding settings for auto-generated ABR renditions**. If you require video and audio to be in the same muxed output, skip this step.

    a. Choose the **Audio 1** tab.

    b. Choose **Remove audio** in the upper right of the **Base encoding settings for auto-generated ABR renditions** section.

12. Optionally, add audio renditions. Follow these steps for each audio rendition that you want in your ABR stack.

    a. From the **Job** pane on the left, choose your output group.

    b. In the **Outputs** section, choose **Add output with captions or audio** to add a new output.

       Output 1 holds your automated ABR video settings and represents every video rendition in your stack.

    c. Choose the new output from the list of outputs.

    d. In the **Encoding settings** section, set up your audio rendition as you would for a job that doesn't use automated ABR. For more information, see Creating audio ABR streaming outputs.

13. Optionally, add captions. Do this as you would for a manually specified ABR stack. For more information, see Setting up captions in AWS Elemental MediaConvert jobs.

14. Optionally, repeat this procedure to create additional ABR packages in different formats. In a job that includes an automated ABR output group, all ABR output groups must use automated ABR.

API, SDK, or CLI

To set up an automated ABR job using the API, SDK, or AWS Command Line Interface (CLI):

If you use the API, CLI, or an SDK, specify the relevant settings in your JSON job specification and then submit it programmatically with your job. For more information about submitting your job programmatically, see one of the introductory topics of the *AWS Elemental MediaConvert API Reference*:

- Getting started with AWS Elemental MediaConvert using the AWS SDKs or the AWS CLI
- Getting started with AWS Elemental MediaConvert using the API

1. Determine the values that you want to set for automated ABR. If you keep all the defaults, you should get good results. For more information, see How automated ABR works.
2. Use the MediaConvert console to generate your JSON job specification. We recommend this approach, because the console functions as an interactive validator against the MediaConvert job schema. Follow these steps to generate your JSON job specification using the console:

    a. Follow the previous procedure for the console.

    b. In the **Job** pane on the left, under **Job settings**, choose **Show job JSON**.

Information for manually editing your JSON job specification

Find additional information, including where each setting belongs in the job settings structure, in the *AWS Elemental MediaConvert API Reference*. Links in this list go to information about the setting in that document:

> **Important**
>
> If you set up automated ABR by manually editing your JSON job specification, rather than exporting it from the MediaConvert console, you must explicitly set `qualityTuningLevel` to `MULTI_PASS_HQ` and `rateControlMode` to `QVBR`.

- Explicitly set these required settings:
    - Set `qualityTuningLevel` to `MULTI_PASS_HQ`.

        - AVC (H.264): [qualityTuningLevel](#) in the H264Settings properties table
        - HEVC (H.265): [qualityTuningLevel](#) in the H265Settings properties table
    - Set `rateControlMode` to `QVBR`.

      When you enable automated ABR, the usual required settings for QVBR, such as `qvbrSettings` and `qvbrQualityLevel`, aren't required. Instead specify the required automated ABR settings.

        - AVC (H.264): [rateControlMode](#) in the H264Settings properties table
        - HEVC (H.265): [rateControlMode](#) in the H265Settings properties table
- Set the accelerated transcoding [mode](#) to `PREFERRED` or `ENABLED`. This is optional, but we strongly recommend it.
- Optionally, specify these limits that relate to the whole ABR stack:
    - **Automated ABR** : [abrSettings](#)
        - **Max renditions**: [maxRenditions](#) in the AutomatedAbrSettings properties table
        - **Max ABR bitrate**: [maxAbrBitrate](#) in the AutomatedAbrSettings properties table
        - **Min ABR bitrate**: [minAbrBitrate](#) in the AutomatedAbrSettings properties table
- Optionally, specify these limits that relate to properties of the renditions in the stack:

  Note

  The API properties that correspond to these MediaConvert console settings function differently depending on whether they are in outputs that are part of an automated ABR stack.

    - **Max resolution**: [width](#) and [height](#)
        - In automated ABR: Use these settings together to represent the maximum possible resolution in the ABR stack.
        - In other outputs: Use these settings together to represent the output resolution.
    - **Max frame rate**: `frameratecontrol`, `numerator`, and `denominator`

      For information about how these properties work and links to them in the API Reference, see [Converting the frame rate of your video](#).

- In automated ABR: Use these settings to specify the frame rate of the highest-bandwidth rendition in your stack. If you don't specify these settings, MediaConvert uses the frame rate of your input video.
- In other outputs: Use these settings to specify the output frame rate.

Automated ABR rules

Use **Automated ABR rules** to specify restrictions for the rendition sizes that MediaConvert creates in your automated ABR stack. We recommend using these rules if your ABR workflow has specific rendition size requirements, but you still want MediaConvert to optimize for video quality and overall file size.

You can define the following rules:

- **Min top rendition size**
- **Min bottom rendition size**
- **Force include renditions**
- **Allowed renditions**

Min top rendition size

Specify a minimum size for the highest video resolution in your ABR stack. The highest resolution is greater than or equal to the value that you enter.

For example: If you specify 1920x1080, the highest resolution in your ABR stack is greater than or equal to 1920x1080.

Min bottom rendition size

Specify a minimum size for the lowest video resolution in your ABR stack. The lowest resolution in your ABR stack is greater than or equal to the value that you enter.

For example: If you specify 512x288, the lowest resolution in your ABR stack is greater than or equal to 512x288.

Force include renditions

Specify one or more video resolutions to include in your ABR stack. To optimize automated ABR, we recommend that you specify as few resolutions as possible.

The ABR stack might include other resolutions that you do not specify here, depending on the **Max renditions** setting. For example: If you specify 2 resolutions under **Force include renditions**, and specify 7 **Max renditions**, then 5 resolutions are automatically determined.

Force include renditions has the following restrictions with other automated ABR rules or settings:

- At least one resolution must be greater than or equal to **Min top rendition size**.
- All resolutions must be greater than or equal to **Min bottom rendition size**.
- **Allowed renditions** cannot be not specified.
- The number of resolutions must be less than or equal to **Max renditions**.
- Duplicate resolutions are ignored.

Allowed renditions

Specify a list of possible video resolutions in your ABR stack. MediaConvert creates an ABR stack exclusively from the list of resolutions that you specify.

Some resolutions in the **Allowed renditions** list might not be included. However you can force a resolution to be included by setting **Required** to **ENABLED**.

Allowed renditions has the following restrictions with other automated ABR rules:

- At least one resolution must be greater than or equal to **Min top rendition size**.
- At least one resolution must be greater than or equal to **Min bottom rendition size**.
- **Force include renditions** cannot be not specified.
- The number of resolutions must be less than or equal to **Max renditions**.
- Duplicate resolutions are ignored.

Automated ABR frequently asked questions

How can I see what renditions AWS Elemental MediaConvert created for me?

You can see the properties of the outputs in your ABR stack in these places:

- The job completion event from Amazon EventBridge. For more information, see Using EventBridge with AWS Elemental MediaConvert.

- The **Job summary** page on the MediaConvert console. For more information, see Viewing your job history.

Will my automated ABR job take a long time to run?

We recommend that you always use accelerated transcoding with automated ABR. When you do, your job should take only slightly longer than an accelerated transcoding job for a manually specified ABR stack with similar outputs. You don't pay more for enabling accelerated transcoding because automated ABR is already billed at the 2 Pass (Quality Optimized) professional tier rate.

When you run an automated ABR job without accelerated transcoding, it takes much longer to run than a job with a manually specified ABR stack with similar outputs..

Why do some of my output renditions have the same resolution?

When display devices stream an ABR asset, they request segments based on the bitrate of the rendition, not based on the resolution of the rendition. Therefore, an ABR stack can have renditions for different bandwidths that have the same resolution. The higher bandwidth rendition will have better quality at the same resolution.

Whether increasing the resolution improves video quality when you go up to the next rendition of the stack depends on the complexity of the video. The ability to automatically adjust these choices on a per-job basis is one of the ways that this feature gives you better results with less effort.

Can I tell ahead of time how many renditions will be in my stack?

No. MediaConvert determines which renditions to use during the transcoding process. Because the encoding decisions depend on qualities of your input video, there's no way to know before running the job what those decisions will be.

You can use the optional limits settings to make sure that the number of renditions, and the size of those renditions, won't exceed what you want.

How will I be billed for an automated ABR stack?

MediaConvert bills you for only the renditions that it writes to your output location. For example, you might set **Max renditions** to 12, but MediaConvert might determine that there is no advantage to creating more than eight renditions. In this case, MediaConvert would bill you for only eight renditions.

Automated ABR is a professional-tier feature and also requires 2 pass encoding. Every rendition is billed per-minute at the 2 Pass (Quality Optimized) rate. For example, say your automated ABR stack ends up with 10 renditions, each of them being 60 minutes long. You would then be charged for 600 minutes. For rates, see AWS Elemental MediaConvert Pricing in the *AWS Cloud Products* website.

What about audio?

Automated ABR does the set up for your video renditions only. You add audio renditions as audio-only outputs inside of your automated ABR output group. For instructions, see Creating an automated ABR stack.

What about captions?

Add captions to your automated ABR package as captions-only output. For instructions, see Creating an automated ABR stack.

Feature restrictions for automated ABR

Jobs that include automated ABR output groups are restricted in the following ways:

- You must use an on-demand queue. You can't use a reserved queue.
- You can enable automated ABR in jobs and job templates only. You can't use automated ABR in output presets.
- In a job that includes an automated ABR output group, all ABR output groups must use automated ABR.
- Your output **Video codec** must be **AVC (H.264)** or **HEVC (H.265)**.
- Your output video **Scaling behavior** must be **Default**, **Stretch to output**, **Fit**, or **Fill**. You can't use **Fit without upscaling**.
- You must specify these required settings when you create your JSON job specification manually. The MediaConvert console sets them for you when you enable automated ABR.
  - Set `qualityTuningLevel` to `MULTI_PASS_HQ`.
  - Set `rateControlMode` to `QVBR`.

Creating Dolby Atmos outputs with AWS Elemental MediaConvert

Dolby Atmos provides an immersive audio experience in cinemas and home theaters. With the right audio input files, you can use AWS Elemental MediaConvert to create Dolby Atmos outputs. You can create streaming outputs that end viewers can experience in their home theaters or file outputs that you can use in your professional workflows.

AWS Elemental MediaConvert can create Dolby Digital Plus with Atmos outputs by either encoding channel-based immersive audio content that you provide as 9.1.6, 7.1.4, or 5.1.4 PCM mono channels, or by passing through already encoded Dolby Digital Plus with Atmos content.

Topics

Using Dolby Atmos passthrough with AWS Elemental MediaConvert

AWS Elemental MediaConvert can create Dolby Digital Plus with Atmos outputs by either encoding audio in 9.1.6, 7.1.4, or 5.1.4 PCM mono channels, or by passing through already encoded Dolby Digital Plus with Atmos content.

You set up your job to pass through Dolby Digital Plus with Atmos content in the same way that you pass through Dolby Digital and Dolby Digital Plus content.

To set up a Dolby Atmos job, passing through finished audio content

1. Open the MediaConvert console at https://console.aws.amazon.com/mediaconvert.
2. Choose **Create job**.
3. Set up your input audio and video as described in Configuring jobs in MediaConvert.
4. Set up your output groups, outputs, and video output selectors as described in Configuring jobs in MediaConvert and Creating outputs. Choose supported containers as listed in Supported output formats.
5. Create audio output selectors as described in Configuring jobs in MediaConvert and Creating outputs.

   Set them up as follows:

a. In the **Job** pane on the left, choose an output that includes audio.
b. In the **Encoding settings** section, choose **Audio 1**.
c. For **Audio codec**, choose **Passthrough**.

Feature Restrictions for Dolby Atmos Passthrough

Note the following restrictions in the AWS Elemental MediaConvert implementation of Dolby Atmos passthrough:

- **Output codec:** You can create Dolby Atmos audio outputs encoded with only the Dolby Digital Plus (EAC3) codec.
- **Output containers:** For file outputs, you can create Dolby Atmos audio in only one of the video containers that supports Dolby Digital Plus: MPEG-4, MPEG-2 Transport Stream, or QuickTime.

Using Dolby Atmos encoding with AWS Elemental MediaConvert

AWS Elemental MediaConvert can encode Dolby Digital Plus with Atmos channel-based, immersive audio, Audio Definition Model Broadcast WAV files, or Dolby Atmos Master File.

> **Note**
>
> Understanding Dolby Atmos is required prerequisite knowledge for using this feature. Your input audio channels must already be set up according the Dolby Atmos standard you are using as input. For more information about Dolby Atmos, see the Dolby online documentation.

Feature Restrictions for Dolby Atmos Encoding

Note the following restrictions in the MediaConvert implementation of Dolby Atmos encoding:

- **Input Channel-Based Immersive (CBI):** MediaConvert supports channel-based immersive (CBI) content.
- **Input Dolby Atmos Master File (DAMF):** MediaConvert supports Dolby Atmos master file (DAMF). This is a collection of 3 files with the extensions, .atmos, .atmos.metadata, and .atmos.audio

- **Input Immersive Audio Bitstream (IAB):** MediaConvert supports immersive audio bistream (IAB).
- **Input Audio Definition Model Broadcast WAV Format (ADM BWF):** MediaConvert supports ADM BWF. It is a single broadcast WAV file contains header data with the .atmos and .atmos.metadata information.
- **Output codec:** You can create Dolby Atmos audio outputs encoded with only the Dolby Digital Plus (EAC3) codec.
- **Output containers:** For file outputs, you can create Dolby Atmos audio in only in one of the video containers that supports Dolby Digital Plus: MPEG-4, MPEG-2 Transport Stream, or QuickTime.
- **Output packages:** For adaptive bitrate (ABR) outputs, you can create Dolby Atmos audio in any of the MediaConvert output group types: CMAF, Apple HLS, DASH ISO, or Microsoft Smooth Streaming.

Setting up a job for Dolby Atmos encoding

AWS Elemental MediaConvert can encode Dolby Digital Plus with Atmos channel-based, immersive audio, Audio Descriptive Model Broadcast WAV files, or Dolby Atmos Master File.

> **Note**
>
> Understanding Dolby Atmos is required prerequisite knowledge for using this feature. Your input audio channels must already be set up according the Dolby Atmos standard you are using as input. For more information about Dolby Atmos, see the Dolby online documentation.

For more details about setting up your job for Dolby Atmos encoding, see one of the following procedures:

[Procedure with separate audio input files]

[Procedure with a single audio input file]

To set up a Dolby Atmos job, with audio inputs as 16 individual .wav files

1. Open the MediaConvert console at https://console.aws.amazon.com/mediaconvert.
2. Choose **Create job**.

3. Set up your input video as described in [Configuring jobs in MediaConvert](#).
4. Set up your input audio selectors as follows:

    a. On the **Create job** page, in the **Job** pane on the left, choose **Input**.

    b. On the right, in the **Audio selectors** section, under **Audio selector 1**, choose **External file**.

    c. For **External file**, provide the path and file name to the .wav file for your first channel. For **Audio selector 1**, this channel must be L.

    > **Important**
    >
    > You must set up the channels in the following order: L, R, C, LFE, Ls, Rs, Lrs, Rrs, Lw, Rw, Ltf, Rtf, Ltm, Rtm, Ltr, Rtr.
    > That is, if your input audio is in separate .wav files, **Audio selector 1** must point to the L channel, **Audio selector 2** must point to the R channel, and so on.

    d. At the top of the **Audio selectors** section, choose **Add audio selector** to create **Audio selector 2**.

    e. Under **Audio selector 2**, choose **External file**.

    f. Specify the path and file name to the .wav file for your second channel. For **Audio selector 2**, this channel must be R.

    g. Repeat the steps to create an audio selector for the rest of your 16 channels. Choose the following channels for each selector:

    - **Audio selector 3**: C
    - **Audio selector 4**: LFE
    - **Audio selector 5**: Ls
    - **Audio selector 6**: Rs
    - **Audio selector 7**: Lrs
    - **Audio selector 8**: Rrs
    - **Audio selector 9**: Lw
    - **Audio selector 10**: Rw
    - **Audio selector 11**: Ltf
    - **Audio selector 12**: Rtf
    - **Audio selector 13**: Ltm

- **Audio selector 14**: Rtm
- **Audio selector 15**: Ltr
- **Audio selector 16**: Rtr

5. Create an input **Audio selector group** as follows:

   a. In the **Audio selector groups** section, choose **Add audio selector group**.

   b. For **Group name**, enter a descriptive name, such as **`Dolby Atmos Audio Group`**.

   c. For **Select audio selectors**, choose each audio selector that you created earlier in this procedure. Choose them in order, starting with **Audio selector 1**.

6. Set up your output groups, outputs, and video output selectors as described in Configuring jobs in MediaConvert and Creating outputs. Choose supported containers as listed in Feature Restrictions for Dolby Atmos Encoding.

7. Create audio output selectors as described in Configuring jobs in MediaConvert and Creating outputs.

   Set them up as follows:

   a. In the **Job** pane on the left, choose an output that includes audio.

   b. In the **Encoding settings** section, choose **Audio 1**.

   c. For **Audio codec**, choose **Dolby Digital Plus JOC (Atmos)**.

      For **Audio source**, choose the audio selector group that you created earlier in this procedure, such as **Dolby Atmos Audio Group**.

   d. For the audio encoding settings, choose values that are suitable for your workflow. For more information, see the Dolby documentation for the Dolby Digital Plus Atmos encoding library.

      Note

      AWS Elemental MediaConvert automatically performs audio normalization on Dolby Digital Plus Atmos outputs. Therefore, there is no **Dialnorm** setting under audio encoding settings.

To set up a Dolby Atmos job, with audio input as a single file with 16 tracks

1. Open the MediaConvert console at https://console.aws.amazon.com/mediaconvert.
2. Choose **Create job**.
3. Set up your input video as described in Configuring jobs in MediaConvert.
4. Set up your input audio selectors as follows:

    a. On the **Create job** page, in the **Job** pane on the left, choose **Input**.

    b. On the right, in the **Audio selectors** section, under **Audio selector 1**, choose **External file**.

    c. For **External file**, provide the path and file name to the .wav file.

    d. For **Selector type**, choose **Track**.

    e. For **Tracks**, list your 16 PCM mono tracks in a comma-separated list. Specify them in the following order: L, R, C, LFE, Ls, Rs, Lrs, Rrs, Lw, Rw, Ltf, Rtf, Ltm, Rtm, Ltr, Rtr.

    - If the tracks of your input audio file are already in that order, then list them that way: **`1, 2, 3, … 16`**.
    - If the tracks of your input audio file are in a different order, list them according to the specified order. For example, if your L channel is in track 3, then list **3** first.

5. Set up your output groups, outputs, and video output selectors as described in Configuring jobs in MediaConvert and Creating outputs. Choose supported containers as listed in Feature Restrictions for Dolby Atmos Encoding.
6. Create audio output selectors as described in Configuring jobs in MediaConvert and Creating outputs.

    Set them up as follows:

    a. In the **Job** pane on the left, choose an output that includes audio.

    b. In the **Encoding settings** section, choose **Audio 1**.

    c. For **Audio codec**, choose **Dolby Digital Plus JOC (Atmos)**.

    For **Audio source**, keep the default **Audio selector 1**.

    d. For the audio encoding settings, choose values that are suitable for your workflow. For more information, see the Dolby documentation for the Dolby Digital Plus Atmos encoding library.

> **Note**
>
> AWS Elemental MediaConvert automatically performs audio normalization on Dolby Digital Plus Atmos outputs. Therefore, there is no **Dialnorm** setting under audio encoding settings.

Creating Dolby Vision outputs with AWS Elemental MediaConvert

Dolby Vision video uses an extended color palette and contrast range with dynamic, per-frame metadata. With AWS Elemental MediaConvert, you can create Profile 5 or Profile 8.1 Dolby Vision outputs from supported sources.

Set up a Dolby Vision job

Use the following steps to set up a Dolby Vision job. For more information about jobs, see the section called "Configuring jobs".

1. For your input file or files, choose from the following:
   - MXF file, with frame-interleaved Dolby Vision metadata or an XML file.
   - IMF package (IMP) with frame-interleaved Dolby Vision metadata or an XML file. Also, specify a composition playlist (CPL) file for your input. If your CPL is from an incomplete IMP, choose **Supplemental IMPs** to specify the location of your supplemental IMPs.
   - Apple ProRes QuickTime MOV, with a Dolby Vision studio metadata XML file.
   - Any input with an HDR10 color space.
   - Any input with an SDR color space.
2. For each output that you want to process with Dolby Vision, do the following:
   1. Make sure that your output settings conform to the limitations listed in Dolby Vision input format support and job setting requirements.
   2. Enable the **Dolby Vision** preprocessor.
   3. Specify a Dolby Vision **Profile** from one of the following choices:
      - **Profile 5**: Includes frame-interleaved Dolby Vision metadata in your output.

- **Profile 8.1**: Includes both frame-interleaved Dolby Vision metadata and HDR10 metadata in your output.

3. Choose an on-demand queue. (Your default queue is on-demand.)

Dolby Vision input format support and job setting requirements

The tables in this section describe Dolby Vision input format support and job setting requirements for implementation with AWS Elemental MediaConvert.

The following table describes input format requirements for Dolby Vision Profile 5 or Profile 8.1 outputs.

Supported inputs with Dolby Vision metadata	Supported inputs without Dolby Vision metadata	Supported output Dolby Vision profile
IMF, MXF • Video codec: JPEG 2000 • Input Dolby Vision metadata: Frame-interleaved or XML file QuickTime (.mov) • Video codec: Apple ProRes • Input Dolby Vision metadata: XML file	HDR10 • Video codec: Any capable of carrying HDR10 SDR • Video codec: Any capable of carrying SDR	Profile 5 Profile 8.1

The following table describes feature limitations and job requirements for Dolby Vision outputs.

Feature	Job setting requirement
Maximum number of input videos or input clips (For Profile 8.1 outputs)	One per job
Input **Frame rate**	All inputs must have the same frame rate. Frame rate conversion is not supported.
Input **Image inserter**	Supported (The brightness of your image will vary along with your video content.)
Output **Frame rate**	**Follow source**
Output **Image inserter**	**Disabled**
Output **Video codec**	**HEVC (H.265)**
Output **Color metadata**	**Insert**
Output video **Resolution (w x h)**	Maximum width: 4096 Maximum height: 4096
Output video codec **Profile**	**Main10/Main** or **Main10/High**
Captions **Destination type**	Burn-in captions are not supported.
Respond to AFD	**None**
Color corrector preprocessor	**Disabled**
Timecode burn-in preprocessor	**Disabled**
Noise reducer preprocessor	**Disabled**
Motion image inserter	**Disabled**

Feature	Job setting requirement
Queue type	**On-demand queue**

Protecting your media assets with encryption and DRM using AWS Elemental MediaConvert

Use encryption to protect your content from unauthorized access. You can use any combination of these encryption options with AWS Elemental MediaConvert:

- Server-side encryption: Keep your output files secure in the Amazon S3 buckets that AWS Elemental MediaConvert reads from and writes to.
- Digital rights management (DRM): Make sure that, once you distribute your content, only authorized viewers can watch it

Topics

Implementing server-side encryption

Server-side encryption with Amazon S3 is one of the encryption options that you can use with AWS Elemental MediaConvert.

You can protect your input and output files at rest by using server-side encryption with Amazon S3:

- To protect your input files, set up server-side encryption as you would for any object in an Amazon S3 bucket. For more information, see Protecting data using server-side encryption in the *Amazon Simple Storage Service User Guide*.
- To protect your output files, specify in your AWS Elemental MediaConvert job that Amazon S3 encrypts your output files as MediaConvert uploads them. By default, your output files are not encrypted. The rest of this topic provides more information about setting up your job to encrypt your output files.

When you set up an AWS Elemental MediaConvert job output for server-side encryption, Amazon S3 encrypts it with a data key. As an additional security measure, the data key itself is encrypted with a master key.

You choose whether Amazon S3 encrypts the data key by using the default Amazon S3 managed key or a KMS key that is managed by AWS Key Management Service (AWS KMS). Using the default Amazon S3 master key is simplest to set up. If you prefer more control over your key, use an AWS KMSkey. For more information about the different types of KMS keys managed with AWS KMS, see [What is AWS Key Management Service?](#) in the *AWS Key Management Service Developer Guide*.

If you choose to use an AWS KMS key, you can specify a customer managed key in your AWS account. Otherwise, AWS KMS uses the AWS managed key for Amazon S3, which has the alias `aws/s3`.

To set up your job outputs for server-side encryption

1. Open the MediaConvert console at https://console.aws.amazon.com/mediaconvert.
2. Choose **Create job**.
3. Set up your input, output groups, and outputs for video and audio, as described in Configuring jobs in MediaConvert and Creating outputs.
4. For each output group that has outputs that you want encrypted, set up server-side encryption:

   a. In the **Job** pane on the left, choose the output group.

   b. In the group settings section on the right, choose **Server-side encryption**. If you use the API or an SDK, you can find this setting in the JSON file of your job. The setting name is `S3EncryptionSettings`.

   c. For **Encryption key management**, choose the AWS service that protects your data key. If you use the API or an SDK, you can find this setting in the JSON file of your job. The setting name is `S3ServerSideEncryptionType`.

      If you choose **Amazon S3**, Amazon S3 encrypts your data key with a customer managed key that Amazon S3 stores securely. If you choose **AWS KMS**, Amazon S3 encrypts your data key with a KMS key that AWS Key Management Service (AWS KMS) stores and manages.

d. If you chose **AWS KMS** in the preceding step, optionally specify the ARN of one of your What is AWS Key Management Service?. If you do, AWS KMS will use that KMS key to encrypt the data key that Amazon S3 uses to encrypt your media files.

   If you don't specify a key for **AWS KMS**, Amazon S3 uses the AWS managed key in your AWS account that is used exclusively for Amazon S3.

e. If you chose **AWS KMS** for **Encryption key management**, grant `kms:Encrypt` and `kms:GenerateDataKey` permissions to your AWS Elemental MediaConvert AWS Identity and Access Management (IAM) role. This allows MediaConvert to encrypt your output files. If you also want to be able to use these outputs as inputs to another MediaConvert job, also grant `kms:Decrypt` permissions. To learn more, see these topics:

   - For more information about setting up an IAM role for AWS Elemental MediaConvert to assume, see Setting up IAM permissions in the Getting Started chapter of this guide.
   - For more information about granting IAM permissions using an inline policy, see the procedure **To embed an inline policy for a user or role** in Adding IAM identity permissions (Console) in the *IAM User Guide*.
   - For examples of IAM policies that grant AWS KMS permissions, including decrypting encrypted content, see Customer managed policy examples in the *AWS Key Management Service Developer Guide*.

5. Run your AWS Elemental MediaConvert job as usual. If you chose **AWS KMS** for **Encryption key management**, remember to grant `kms:Decrypt` permissions to any user or role that you want to be able to access your outputs.

Digital rights management (DRM)

DRM is one of the encryption options that you can use with AWS Elemental MediaConvert.

Protect your content from unauthorized use through encryption. Digital rights management (DRM) systems provide keys to AWS Elemental MediaConvert for content encryption, and licenses to supported players and other consumers for decryption.

> **Note**
>
> To encrypt content, you must have a DRM solution provider.

- For an overview, see https://docs.aws.amazon.com/speke/latest/documentation/what-is-speke.html#services-architecture.
- To get set up, see https://docs.aws.amazon.com/speke/latest/documentation/customer-onboarding.html.

The only exception to this requirement is with the Apple HLS streaming protocol, where you can choose to either define your own static keys or use a DRM provider.

Topics

- Encrypting content
- Using encrypted content keys with DRM
- Troubleshooting DRM encryption

Encrypting content

Use the following procedure to enable content encryption in **CMAF**, **DASH ISO**, **Apple HLS**, and **MS Smooth** output groups.

AWS Elemental MediaConvert supports the following combinations of output groups and DRM systems, with SPEKE v1.0.

Output Group	Microsoft PlayReady	Google Widevine	Apple FairPlay
DASH	√	√	
Apple HLS			√
Microsoft Smooth	√		
CMAF	√	√	√

To use this procedure, you should be comfortable working with output groups. For more information, see the section called "Step 3: Create output groups".

To encrypt content

1. Set up your transcoding job as usual. For more information, see the section called “Configuring jobs”.
2. On the **Create job** page, in the **Job** pane on the left, under **Output groups**, choose an output group that you want to enable encryption for.
3. Turn on **DRM encryption**.
4. For **CMAF** and **Apple HLS** output groups, choose the encryption method. Make sure that you choose an encryption method that works with the DRM system that you use.

   For **DASH ISO** and **MS Smooth** output groups, you don't specify the encryption method. MediaConvert always uses AES-CTR (AES-128) encryption with these output groups.
5. For **CMAF** and **Apple HLS** output groups, choose the source for the content encryption key. For **Key provider type**, choose **SPEKE** to encrypt using a key provided by your DRM solution provider, or choose **Static key** to enter your own key.

   For **DASH ISO** and **MS Smooth** output groups, you don't specify the source for the content encryption key. With these output groups, MediaConvert does DRM only with a SPEKE-compliant key provider.

   - For **SPEKE**, fill in the encryption parameter fields. For more information, see the section called “SPEKE encryption parameters”.
   - For **Static Key**, see the section called “Static key encryption parameters”.

> **Note**
>
> MediaConvert does not support DRM content encryption for the following output video codec: AV1.

SPEKE encryption parameters

When you request encryption, you provide input parameters that allow the service to locate your DRM solution provider's key server, to authenticate you as a user and to request the proper encoding keys. Some options are available only for particular output groups.

Enter the SPEKE encryption parameters as follows:

- For **Resource ID**, enter an identifier for the content. The service sends this to the key server to identify the current endpoint. How unique you make this depends on how fine-grained you want access controls to be. The service does not allow you to use the same ID for two simultaneous encryption processes. The resource ID is also known as the content ID.

  The following example shows a resource ID.

  ```
  MovieNight20171126093045
  ```

- For **System ID**, enter unique identifiers for your streaming protocol and DRM system. The number of system IDs that you can specify varies depending on the output group type:
  - CMAF – For **System IDs signaled in DASH**, specify at least one and up to three IDs. For **System ID signaled in HLS**, specify one ID.
  - DASH – For **System ID**, specify at least one and up to two IDs.
  - Apple HLS – For **System ID**, specify one ID.

  If you provide more than one system ID in a single field, enter them on separate lines, and don't separate them with commas or any other punctuation.

  For a list of common system IDs, see DASH-IF System IDs. If you don't know your IDs, request them from your DRM solution provider.

- For **URL**, enter the URL of the API Gateway proxy that you set up to talk to your key server. The API Gateway proxy must reside in the same AWS Region as MediaConvert.

  The following example shows a URL.

  ```
  https://1wm2dx1f33.execute-api.us-west-2.amazonaws.com/SpekeSample/copyProtection
  ```

- (Optional) For **Certificate ARN**, enter a 2048 RSA certificate ARN to use for content key encryption. Use this option only if your DRM key provider supports content key encryption. If you use this and your key provider doesn't support it, the request fails.

  To enter a certificate ARN here, you must have already imported the corresponding certificate into AWS Certificate Manager, entered the certificate ARN from ACM into the MediaConvert **Certificates** pane, and associated it with MediaConvert. For more information, see the section called "Using encrypted content keys with DRM".

  The following example shows a certificate ARN.

```
arn:aws:acm:region:123456789012:certificate/97b4deb6-8983-4e39-918e-ef1378924e1e
```

Additional configuration options for Apple HLS and CMAF

- (Optional) For **Constant initialization vector** enter a 128-bit, 16-byte hex value represented by a 32-character string, to be used with the key for encrypting content.

Static key encryption parameters

The following options are for static key encryption:

- **Static key value –** A valid string for encrypting content.
- **URL –** The URL to include in the manifest so that player devices can decrypt the content.

Using encrypted content keys with DRM

For the most secure DRM encryption solution, use encrypted content keys in addition to encrypted content. To use encrypted content keys, you must import suitable certificates into AWS Certificate Manager (ACM), and then prepare them for use with AWS Elemental MediaConvert. For information about ACM, see the AWS Certificate Manager User Guide.

Run AWS Certificate Manager in the same Region as you run AWS Elemental MediaConvert.

To prepare a certificate for DRM content key encryption

1. Obtain a 2048 RSA, SHA-512-signed certificate.
2. Open the ACM console at https://console.aws.amazon.com/acm/.
3. Import the certificate into ACM according to the instructions at Importing certificates into AWS Certificate Manager. Note the resulting certificate ARN because you will need it later.

   For use in DRM encryption, your certificate must have a status of **Issued** in ACM.
4. Open the MediaConvert console at https://console.aws.amazon.com/mediaconvert/.
5. In the navigation pane, under **Certificates**, enter your certificate ARN, and then choose **Associate certificate**.

To find certificates that are associated with AWS Elemental MediaConvert

In the ACM console, list and display your certificates to find the ones that you have associated with MediaConvert. In the **Details** section of the certificate's description, you can see the MediaConvert association and retrieve the certificate ARN. For more information, see List ACM–managed certificates and Describe ACM certificates.

To use a certificate in AWS Elemental MediaConvert

When you use DRM encryption, provide one of your associated certificate ARNs in the SPEKE encryption parameters. This enables content key encryption. You can use the same certificate ARN for multiple jobs. For information, see the section called “Encrypting content”.

To renew a certificate

To renew a certificate that you have associated with AWS Elemental MediaConvert, reimport it in AWS Certificate Manager. The certificate renews without any disruption of its use in MediaConvert.

To delete a certificate

To delete a certificate from AWS Certificate Manager, you must first dissociate it from any other service. To dissociate a certificate from AWS Elemental MediaConvert, copy its certificate ARN from ACM, navigate to the MediaConvert **Certificates** pane, enter the certificate ARN, and then choose **Dissociate certificate**.

Troubleshooting DRM encryption

If the DRM system key server is unavailable when AWS Elemental MediaConvert requests keys, the console displays the following message: Key Server Unavailable.

Content key encryption adds another layer of complexity to your jobs. If you run into problems on a job that has content key encryption enabled, remove the certificate ARN from your job settings and troubleshoot the job using clear key delivery. When you have that working, reenter the certificate ARN, and then try the job again.

If you contact the AWS Support Center for troubleshooting purposes, have the following information available:

- Region that the job was run in
- Job ID
- Account ID
- Name of your DRM solution provider
- Any other details about the problem that you are having that might assist with troubleshooting

Working with video frame rates in AWS Elemental MediaConvert

The *frame rate* of a video asset represents how quickly video player devices play back the frames of a video, in frames per second (fps). For example, films have a frame rate of 24 fps, NTSC television broadcasts are 29.97/59.94 fps, and PAL television broadcasts are 50/25 fps. If you keep the MediaConvert default settings in your job, your output video will have the same frame rate as your input video.

Some videos have a frame rate that varies over the duration of the video. For example, some cameras automatically generate video that uses more frames for high-action sequences and fewer frames for sequences with less motion. MediaConvert supports variable frame rate (VFR) inputs, but creates only constant frame rate (CFR) outputs. For more information, see Using variable frame rate inputs.

Topics

Settings for frame rate conversion

To create outputs that have a different frame rate than your input, use these MediaConvert settings:

Frame rate (`framerateControl`, `framerateNumerator`, `framerateDenominator`)

> For frame rate conversion, specify a frame rate in your output encoding settings that's different from your input video frame rate. MediaConvert will then create an output that has the frame rate you specify, rather than the frame rate of your input video.
>
> Specifying your output frame rate directly in your JSON job specification can be more complex than doing so in the MediaConvert console. For details, see the procedure for using the API, CLI, and SDK in the topic Converting the frame rate of your video.

Frame rate conversion algorithm (`framerateConversionAlgorithm`)

Choose how you want MediaConvert to increase or decrease the frame rate. The best choice for this setting depends on the content of your video.

When you use **Drop duplicate**, MediaConvert copies or deletes frames but doesn't alter them. This preserves the picture quality of each individual frame, but might introduce stuttering in some conversions. For numerically simple conversions, such as 60 fps to 30 fps, Drop duplicate is often the best choice.

When you use **Interpolate**, MediaConvert blends frames together to avoid the need to repeat or remove frames. This results in smooth motion, but might introduce undesirable video artifacts. For numerically complex conversions, Interpolate is likely to provide better results than Drop duplicate.

When you use **FrameFormer**, MediaConvert uses the InSync FrameFormer library. The conversion uses motion-compensated interpolation based on the content of your input video. FrameFormer performs various frame rate conversion techniques on a scene-by-scene basis and can use different techniques on different regions of each frame. FrameFormer does the conversion based on automatic detection of your source video's underlying cadence, rather than relying on the frame rate reported in the file's metadata.

Feature limitations:

- You can use FrameFormer with inputs that have resolutions up to 4K only. MediaConvert doesn't support FrameFormer conversion with 8K inputs.
- You can use FrameFormer only with jobs that you run through an on-demand queue. You can't use reserved queues with FrameFormer.

Using FrameFormer increases the transcoding time and incurs a significant add-on cost. For more information, see the [MediaConvert pricing page](https://aws.amazon.com/mediaconvert/pricing/).

Depending on the conversion, you might also use these settings:

Slow PAL (`slowPal`)

When you convert the frame rate from 23.976 or 24 frames per second (fps) to 25 fps, you can optionally enable **Slow PAL** (slow phase alternating line). When you enable slow PAL, instead of duplicating frames to increase the frame rate, MediaConvert relabels the video frames as

25 fps and resamples your audio to keep it synchronized with the video. Slow PAL frame rate conversion slightly reduces the duration of the video. Generally, you use slow PAL to convert a cinema format for file-based playback or internet streaming.

Telecine (`telecine`)

When you convert the frame rate from 23.976 frames per second (fps) to 29.97 fps, and your output scan type is interlaced, you can optionally set **Telecine** to **Hard** or **Soft** to create a smoother picture. Generally, you use telecine when you're preparing a video asset for broadcasting to set-top boxes.

For more information, see Telecine.

Scan type (`inputScanType`)

Use this setting only with progressive segmented frame (PsF) inputs. MediaConvert automatically detects progressive and interlaced inputs. But it doesn't detect PsF. When your input is PsF, set **Scan type** to **PsF** for better preservation of quality when you do deinterlacing and frame rate conversion.

Using variable frame rate inputs in AWS Elemental MediaConvert

Some videos have a frame rate that varies over the duration of the video. Some cameras—for example, the cameras in many smartphones—automatically generate video that uses more frames for high-action sequences and fewer frames for sequences with less motion. MediaConvert supports variable frame rate (VFR) inputs, but creates only constant frame rate (CFR) outputs.

The default setting for output frame rate is **Follow source**. **Follow source** causes different behavior depending on whether your input video has a constant or variable frame rate.

- For constant frame rate inputs, **Follow source** results in outputs that have the same frame rate as the input video.
- For variable frame rate inputs, **Follow source** results in outputs that have a constant frame rate output, with a frame rate that is the average of the input frame rates, rounded up to the nearest whole number standard frame rate: 1, 5, 10, 15, 24, 30, 50, or 60 fps.

Feature restrictions

MediaConvert support for variable frame rate video is limited in these ways:

- Variable frame rates are supported as input only. Outputs are only constant frame rate.
- Variable frame rate inputs are supported in these containers only: MP4, MOV, WEBM, and MKV.

Converting the frame rate of your video

After you know how you want to specify the relevant settings, use one of the following procedures to set up your job. For conceptual information and guidance about choosing the right values for these settings, see Settings for frame rate conversion.

Console

To set up your transcoding job with frame rate conversion using the MediaConvert console:

1. Determine the values that you want to set for frame rate conversion. For more information, see Settings for frame rate conversion.
2. Set up your job inputs and outputs as described in Configuring jobs in MediaConvert.
3. On the **Create job** page, in the **Job** pane on the left, choose the output that you want to use frame rate conversion with.

   > **Tip**
   >
   > To find a specific encoding setting on the MediaConvert console, use your browser's search function.

4. In the **Encoding settings** section, for **Frame rate**, choose the frame rate that you want for your output. If the frame rate that you want isn't listed, choose **Custom**. Then specify your frame rate as a fraction in the fields to the right of **Frame rate**.
5. For **Framerate conversion algorithm**, choose the algorithm most suited to your content.

   Framerate conversion algorithm isn't available on the MediaConvert console until you choose your output frame rate.
6. Optional. If your output is 25 fps and you want to use **Slow PAL**, enable it.
7. Optional. If you're converting from 23.976 fps to 29.97 fps and you want to do a telecine conversion, set **Telecine** to **Hard** or **Soft**.

   Telecine isn't available on the MediaConvert console until you set your output frame rate to 29.97.

API, SDK, or CLI

If you use the API, CLI, or an SDK, specify the relevant settings in your JSON job specification and then submit it programmatically with your job. For more information about submitting your job programmatically, see one of the introductory topics of the *AWS Elemental MediaConvert API Reference*:

- Getting started with AWS Elemental MediaConvert using the AWS SDKs or the AWS CLI
- Getting started with AWS Elemental MediaConvert using the API

To set up your transcoding job with frame rate conversion using the API, SDK or AWS Command Line Interface (CLI):

1. Determine the values you want to set for frame rate conversion. For more information, see Settings for frame rate conversion.
2. Use the MediaConvert console to generate your JSON job specification. We recommend this approach, because the console functions as an interactive validator against the MediaConvert job schema. Follow these steps to generate your JSON job specification using the console:

    a. Follow the previous procedure for the console.

    b. In the **Job** pane on the left, under **Job settings**, choose **Show job JSON**.

Find additional information, including where each setting belongs in the job settings structure, in the *AWS Elemental MediaConvert API Reference*. Links in this list go to information about the setting in that document:

- **Frame rate control** (`framerateControl`)

  Use the frame rate control setting to specify whether MediaConvert uses the frame rate of your input sources or the frame rate that you specify with the `framerateNumerator` and `framerateDenominator` settings.

> **Note**
>
> The default behavior for this setting is to follow source. Therefore, if you keep this setting out of your JSON job specification, MediaConvert ignores any values you provide for `framerateNumerator` and `framerateDenominator`.

- AV1: `framerateControl`
- AVC (H.264): `framerateControl`
- HEVC (H.265): `framerateControl`
- MPEG-2: `framerateControl`
- Apple ProRes: `framerateControl`
- VP8: `framerateControl`
- VP9: `framerateControl`

• **Frame rate** (`framerateNumerator` and `framerateDenominator`)

  In the MediaConvert job settings schema, frame rate is represented as a fraction, to retain precision with irrational numbers. Therefore, specify your frame rate value as `framerateNumerator` divided by `framerateDenominator`. For values for common frame rates, see the table following this list of settings.

  Links to `framerateNumerator`

  - AV1: `framerateNumerator`
  - AVC (H.264): `framerateNumerator`
  - HEVC (H.265): `framerateNumerator`
  - MPEG-2: `framerateNumerator`
  - Apple ProRes: `framerateNumerator`
  - VP8: `framerateNumerator`
  - VP9: `framerateNumerator`

  Links to `framerateDenominator`

  - AV1: `framerateDenominator`
  - AVC (H.264): `framerateDenominator`

    - HEVC (H.265): framerateDenominator
    - MPEG-2: framerateDenominator
    - Apple ProRes: framerateDenominator
    - VP8: framerateDenominator
    - VP9: framerateDenominator
- **Frame rate conversion algorithm** (`framerateConversionAlgorithm`)
    - AV1: framerateConversionAlgorithm
    - AVC (H.264): framerateConversionAlgorithm
    - HEVC (H.265): framerateConversionAlgorithm
    - MPEG-2: framerateConversionAlgorithm
    - Apple ProRes: framerateConversionAlgorithm
    - VP8: framerateConversionAlgorithm
    - VP9: framerateConversionAlgorithm
- **Slow PAL** (`slowPal`)
    - AVC (H.264): slowPal
    - HEVC (H.265): slowPal
    - MPEG-2: slowPal
    - Apple ProRes: slowPal
- **Telecine** (`telecine`)
    - AVC (H.264): telecine
    - HEVC (H.265): telecine
    - MPEG-2: telecine
    - Apple ProRes: telecine
    - **Scan type** (InputScanType)

Common frame rate ratios

Frame rate common name	Value for framerate Numerator	Value for framerate Denominator
23.976	24,000	1,001

Frame rate common name	Value for framerate Numerator	Value for framerate Denominator
29.97	30,000	1,001
59.94	60,000	1,001

Working with progressive and interlaced scan types in AWS Elemental MediaConvert

Progressive and *interlaced* are two types of video display methods. Modern display devices detect whether a video is interlaced or progressive and automatically play back the video correctly. But, progressive video looks much better on modern screens.

To get the best results with using interlacing/deinterlacing and converting to and from telecine, you must consider how your input video was recorded and what transformations have been done to it. For example, when you apply deinterlacing to an input that is not interlaced, your output video quality suffers.

Topics

- Basic scan type vocabulary
- Settings for scan type conversion
- Valid settings combinations
- Converting the scan type of your video

Basic scan type vocabulary

Progressive video

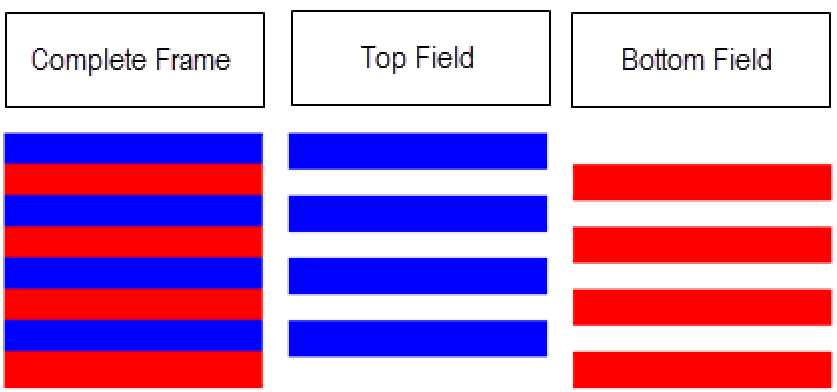
> *Progressive video* includes all lines in all frames. It looks better on modern screens because it drastically reduces the amount of image flicker the viewer sees on the screen. Devices displaying progressive video will re-draw all horizontal lines in a frame. For example, a device running at 50 Hertz playing a 1080 progressive video re-draws 1080 lines (every line in the frame) 50 times per second.

Interlaced video

Interlaced video uses a technique that doubles the perceived frame rate of a video display without consuming extra bandwidth. On older displays, most people won't notice decreased video quality with interlaced video. Devices that support interlaced video re-draw every *other* horizontal line in a frame. For example, a device running at 50 Hertz playing a 1080 interlaced video redraws 540 lines (half of the lines in the frame) 50 times per second.

Field polarity for interlaced frames

Interlaced video contains two fields of a video frame, each one made up of every other horizontal line the image. *Field polarity* in video distinguishes between these two sets of lines. A set's polarity indicates whether the top field comes first or bottom field comes first. In the following illustration, the set with top field polarity is shown in blue and contains the topmost line. The set with bottom field polarity is shown in red and contains the second horizontal line from the top. The complete frame contains both, with each set being refreshed alternately.



When you create interlaced outputs with MediaConvert, you can specify which field polarity comes first with the setting **Interlace mode**.

Settings for scan type conversion

To convert between interlaced to progressive video, specify the MediaConvert settings covered in this topic. This topic offers conceptual information and guidance for choosing values for the MediaConvert settings related to interlacing and deinterlacing. For directions for specifying them, see the procedures in the topic Converting scan type.

Valid values for some of these settings depend on what you choose for the other settings. For a table that shows how to specify them together correctly, see Valid settings combinations.

Deinterlacer preprocessor (`Deinterlacer`)

Use this parent setting to enable and disable deinterlacing. If you simply enable the deinterlacer without specifying any further deinterlacing settings, your job will convert interlaced content to progressive. For the default deinterlacing to work correctly, your input video must be interlaced and the frames of your input video must not have metadata that tags them as progressive.

Deinterlace Control (`DeinterlacerControl`)

This setting is a child of the deinterlacer setting. You can optionally use **Deinterlace control** to have MediaConvert deinterlace all frames of your input video, including those that are tagged as progressive. Only use this setting when you know that this metadata in your input video is wrong.

Deinterlace algorithm (`DeinterlaceAlgorithm`)

This setting is a child of the deinterlacer setting. You can optionally use **Deinterlace algorithm** to specify the way that MediaConvert does the deinterlacing to get the best quality for your content. For sharper pictures, choose one of the motion adaptive interpolation options (**Interpolate** or **Interpolate ticker**). For smoother motion, choose one of the blend options (**Blend** or **Blend ticker**). When your source file includes moving text, such as a scrolling headline at the bottom of the frame, choose the ticker version of the algorithm.

Deinterlace mode (`DeinterlacerMode`)

This setting is a child of the deinterlacer setting. You can optionally use **Deinterlace mode** to modify how MediaConvert applies deinterlacing.

Keep the default value, **Deinterlace**, to do regular deinterlacing.

Choose **Inverse telecine** to convert hard telecine (29.97 fps, interlaced) to progressive video at 23.976 fps. When you use inverse telecine, you must still specify your output frame rate as 23.97. MediaConvert doesn't automatically set this.

Choose **Adaptive** to have MediaConvert automatically detect interlaced inputs and apply deinterlacing and inverse telecine to them. Adaptive deinterlace mode is useful when you use output presets, job templates, or custom programming to apply the same job settings to transcode an entire library of assets.

> **Note**
>
> When you choose **Adaptive** for this setting, MediaConvert automatically uses inverse telecine as well.

Interlace mode (`interlaceMode`)

When you create interlaced video, from either progressive or interlaced inputs, use this MediaConvert setting. The default value of this setting is **Progressive**, so you can ignore this setting unless you want an interlaced output.

> **Note**
>
> When you use an interlaced input and you keep the default setting, **Progressive**, for **Interlace mode**, you should also enable **Deinterlace**. Otherwise, your progressive output will have very poor video quality.

When you create interlaced outputs, use **Interlace mode** to specify the field polarity of your outputs. You can either directly specify the field that comes first, or you can set it to follow the polarity of the source input. For jobs that have multiple inputs, the output might have a mix of top and bottom field first, depending on the polarity of the inputs.

When you set **Interlace mode** to follow the source and your input is progressive, the output's field polarity depends on which of the follow options you set. **Follow, top field** results in an output that's top field first. **Follow, bottom field** results in an output that's bottom field first.

Scan type (`inputScanType`)

Use this setting only when your input is progressive segmented frame (PsF). MediaConvert automatically detects progressive and interlaced inputs. But it doesn't detect PsF. When your input is PsF, set **Scan type** to **PsF** for better preservation of quality when you do deinterlacing and frame rate conversion.

Valid settings combinations

Use this table to confirm that the scan type settings you intend to use are valid together and that they work with the scan type of your source input.

Note

Deinterlace algorithm doesn't appear in this table, because whenever it makes sense to enable **Deinterlacer**, you can choose any value regardless of your other settings.

To convert this input	To this output	Use these settings values
Progressive	Progressive	**Deinterlacer**: Disabled **Deinterlace control**: N/A **Deinterlace mode**: N/A **Interlace mode**: Progressive (default) **Telecine**: None (default)
Interlaced	Progressive	**Deinterlacer**: Enabled **Deinterlace control**: Either value **Deinterlace mode**: Deinterlace (default) **Interlace mode**: Progressive (default) **Telecine**: None (default)
Hard telecine	Progressive	**Deinterlacer**: Enabled **Deinterlace control**: Either value

To convert this input	To this output	Use these settings values
		Deinterlace mode: Inverse telecine **Interlace mode**: Progressive (default) **Telecine**: None (default) **Frame rate**: 23.976
Hard telecine	Progressive *When you want to use **Adaptive** for **Deinterlace mode***	**Deinterlacer**: Enabled **Deinterlace control**: Normal **Deinterlace mode**: Adaptive **Interlace mode**: Progressive (default) **Telecine**: None (default) **Frame rate**: 23.976
Soft telecine	Progressive	**Deinterlacer**: Disabled **Deinterlace control**: N/A **Deinterlace mode**: N/A **Interlace mode**: Progressive (default) **Telecine**: None (default) **Frame rate**: 23.976

To convert this input	To this output	Use these settings values
Multiple inputs, some interlaced and some progressive	Progressive	**Deinterlacer**: Enabled **Deinterlace control**: Normal **Deinterlace mode**: Adaptive **Interlace mode**: Progressive (default) **Telecine**: None (default)
Progressive	Hard telecine	**Deinterlacer**: Disabled **Deinterlace control**: N/A **Deinterlace mode**: N/A **Interlace mode**: Any value except progressive **Telecine**: Hard **Frame rate**: 29.97
Hard telecine	Hard telecine	**Deinterlacer**: Disabled **Deinterlace control**: N/A **Deinterlace mode**: N/A **Interlace mode**: Any value except progressive **Telecine**: None **Frame rate**: Follow source

To convert this input	To this output	Use these settings values
Soft telecine	Hard telecine	**Deinterlacer**: Disabled **Interlace mode**: Any value except progressive **Telecine**: Hard **Framerate**: 29.97
Multiple inputs, some interlaced and some progressive	Hard telecine	**Deinterlacer**: Disabled **Deinterlace control**: N/A **Deinterlace mode**: N/A **Interlace mode**: Any value except progressive **Telecine**: Hard **Framerate**: 29.97
Interlaced	Interlaced	**Deinterlacer**: Disabled **Deinterlace control**: N/A **Deinterlace mode**: N/A **Interlace mode**: Any value except progressive **Telecine**: None

To convert this input	To this output	Use these settings values
Multiple inputs, some interlaced and some progressive	Interlaced	**Deinterlacer**: Disabled **Deinterlace control**: N/A **Deinterlace mode**: N/A **Interlace mode**: Any value except progressive **Telecine**: None
Progressive	Soft telecine	**Deinterlacer**: Disabled **Deinterlace control**: N/A **Deinterlace mode**: N/A **Interlace mode**: Any value except progressive **Telecine**: Soft
Hard telecine	Soft telecine	**Deinterlacer**: Enabled **Deinterlace control**: Either value **Deinterlace mode**: Inverse telecine **Interlace mode**: Any value except progressive **Telecine**: Soft **Framerate**: 23.967

To convert this input	To this output	Use these settings values
Hard telecine	Soft telecine *When you want to use **Adaptive** for **Deinterlace mode***	**Deinterlacer**: Enabled **Deinterlace control**: Normal **Deinterlace mode**: Adaptive **Interlace mode**: Any value except progressive **Telecine**: Soft **Framerate**: 23.967
Soft telecine	Soft telecine	**Deinterlacer**: Disabled **Deinterlace control**: N/A **Deinterlace mode**: N/A **Interlace mode**: Any value except progressive **Telecine**: Soft
Multiple inputs, some interlaced and some progressive	Soft telecine	**Deinterlacer**: Enabled **Deinterlace control**: Normal **Deinterlace mode**: Adaptive **Interlace mode**: Any value except progressive **Telecine**: Soft **Framerate**: 23.967

Converting the scan type of your video

After you know how you want to specify the relevant settings, use one of the following procedures to set up your job. For conceptual information and guidance about choosing the right values for these settings, see Settings for scan type conversion.

To set up your transcoding job to convert scan type and telecine (console)

1. Consult the topic Settings for scan type conversion to determine the values that you want to set for interlacing or deinterlacing.
2. Set up your job inputs and outputs as described in Configuring jobs in MediaConvert.
3. On the **Create job** page, in the **Job** pane on the left, choose the output that you want to work with.
4. Find the settings you need in the **Encoding settings** section as follows:

   - **Deinterlacer** preprocessor: Choose **Deinterlacer** from the list of preprocessors at the bottom of the **Encoding settings** section.
   - **Deinterlace Control**: Find this setting in the **Deinterlacer** section after you enable the deinterlacer.
   - **Deinterlace algorithm**: Find this setting in the **Deinterlacer** section after you enable the deinterlacer.
   - **Deinterlace mode**: Find this setting in the **Deinterlacer** section after you enable the deinterlacer.
   - **Interlace mode**: Find this setting directly under **Encoding settings**. You might want to use your web browser's search function to find this setting.
   - **Telecine**: This setting is only visible in the MediaConvert console when you set **Frame rate** to **29.970**. Find **Frame rate** directly under **Encoding settings**. You might want to use your web browser's search function to find this setting.

     The default value for **Telecine** is **None**. Therefore, you only need to make this setting visible in the MediaConvert console when you are creating a telecine output.

To set up your transcoding job to convert scan type and telecine (API, CLI, or SDK)

If you use the API, CLI, or an SDK, specify the relevant settings in your JSON job specification and then submit it programmatically with your job. For more information about submitting your job programmatically, see one of the introductory topics of the *AWS Elemental MediaConvert API Reference*:

- Getting started with AWS Elemental MediaConvert using the AWS SDKs or the AWS CLI
- Getting started with AWS Elemental MediaConvert using the API

1. Consult the topic Settings for scan type conversion to determine the values that you want to set for interlacing or deinterlacing.
2. Use the MediaConvert console to generate your JSON job specification. We recommend this approach, because the console functions as an interactive validator against the MediaConvert job schema. Follow these steps to generate your JSON job specification using the console:

    a. Follow the previous procedure for the console.

    b. In the **Job** pane on the left, under **Job settings**, choose **Show job JSON**.

   Find additional information, including where each setting belongs in the job settings structure, in the *AWS Elemental MediaConvert API Reference*. Links in this list go to information about the setting in that document:

    - **Deinterlacer** preprocessor: `Deinterlacer`
    - **Deinterlace Control**: `DeinterlacerControl`
    - **Deinterlace algorithm**: `DeinterlaceAlgorithm`
    - **Deinterlace mode**: `DeinterlacerMode`
    - **Interlace mode** (`interlaceMode`)
        - AVC (H.264): `interlaceMode`
        - HEVC (H.265): `interlaceMode`
        - MPEG-2: `interlaceMode`
        - Apple ProRes: `interlaceMode`
    - **Telecine** (`telecine`)
        - AVC (H.264): `telecine`
        - HEVC (H.265): `telecine`
        - MPEG-2: `telecine`
        - Apple ProRes: `telecine`

- **Scan type** (`InputScanType`)

Working with telecine in AWS Elemental MediaConvert

> **Tip**
>
> If your video processing workflow doesn't require telecine, choose a standard progressive output.

When you convert the frame rate from 23.976 frames per second (fps) to 29.97 fps, and your output scan type is interlaced, you can optionally use the **Telecine** setting to enable hard or soft telecine. *Hard telecine* produces a 29.97i output by duplicating interlaced video frame fields onto multiple frames. *Soft telecine* produces a 23.976 fps output that signals to the video player device to do the conversion during play back. Generally, broadcasters use telecine when preparing film assets for broadcasting to NTSC set-top boxes.

How you set some of the related MediaConvert settings depends on the scan type and frame rate of your input, and whether your input has telecine frames already. Details about related settings are in the following list. For a summary table, see Valid settings combinations.

Related settings

Frame rate (`framerateControl`, `framerateNumerator`, `framerateDenominator`)

When you enable the telecine setting, you must still specify your output frame rate as 29.97. MediaConvert doesn't automatically set this.

Frame rate conversion algorithm (`framerateConversionAlgorithm`)

When your input frame rate isn't 23.976 and you enable **Telecine**, MediaConvert uses the frame rate conversion algorithm that you specify to convert your frame rate to 23.976 before applying telecine. When your input frame rate is 23.976, MediaConvert ignores any value you set for **Frame rate conversion algorithm**.

Deinterlacer preprocessor (`Deinterlacer`)

Enable the deinterlacer when you want to remove interlacing to create a progressive output. When you remove interlacing from a telecine input, make sure to also set **Deinterlace mode** to **Inverse telecine**.

For more information about deinterlacing, see Progressive and interlaced scan types.

Interlace mode (`interlaceMode`)

When your input is progressive and you set up a telecine output, apply interlacing using **Interlace mode**.

For more information about interlacing, see Progressive and interlaced scan types.

Scan type (`inputScanType`)

Use this setting only when your input is progressive segmented frame (PsF). MediaConvert automatically detects progressive and interlaced inputs. But it doesn't detect PsF. When your input is PsF, set **Scan type** to **PsF** for better preservation of quality when you do deinterlacing and frame rate conversion.

For MediaConvert console and API procedures for setting up a job to convert a video to or from telecine, see Converting scan type.

Creating HDR outputs with AWS Elemental MediaConvert

You can create HDR content with AWS Elemental MediaConvert in the following ways:

Passing through HDR content

You can pass through HDR content by using an HDR input and creating outputs in the same format, with the same metadata. To do that, you keep the color space default settings, choose HEVC for your codec, and choose a 10-bit profile. MediaConvert automatically reads the HDR metadata, including color space, from the video source. For detailed instructions, see the section called "Passing through HDR content".

Correcting inaccurate or missing HDR metadata

To provide HDR 10 metadata that is not present in your input, or to correct metadata that is wrong, add it or overwrite it in the input video settings. This doesn't change the video content and is different from the color space conversion that you can do in your output video settings. For detailed instructions, see the section called "Replacing inaccurate or missing HDR metadata".

Converting from an hdr format to a different HDR format

You can convert your input color space to a different output color space. You do that by choosing the output color space in the output **Color corrector** settings. For detailed instructions, see the section called “Converting the color space”.

Changing SDR input to HDR format

If your input is SDR, you can convert the color space to an HDR format. This process creates output that is formatted as HDR and automatically converts the metadata to match. You do that by choosing the output color space in the output **Color corrector** settings. For detailed instructions, see the section called “Converting the color space”.

> **Note**
>
> This process doesn't upgrade the dynamic range of the video content. These outputs will play on HDR player devices, and they will appear generally brighter than the original SDR content. But the results are not the same as content that has been remastered from SDR to HDR by a color grader.

Changing HDR input to SDR format

If your input is HDR, you can convert the format to any supported SDR color space.

> **Note**
>
> When professional color graders convert an asset from HDR to SDR, they make artistic decisions about where to map colors from the larger space that don't exist in the smaller space. There is no standard formula to map these values automatically. The tone mapping technology that MediaConvert uses to do automatic conversion from HDR to SDR approximates the outcome of manually regrading from HDR to SDR. This automatic conversion works well with most content, but we recommend that you review your outputs to confirm the tone mapping results.

Topics

- HDR support in AWS Elemental MediaConvert
- Passing through HDR content
- Replacing inaccurate or missing HDR metadata

- Converting the color space

HDR support in AWS Elemental MediaConvert

AWS Elemental MediaConvert supports HDR with HEVC video assets in the following outputs: MPEG2-TS, DASH, and CMAF.

You can set up your CMAF outputs to be compatible with Apple HLS player devices. For more information, see the section called "Creating HDR HLS outputs that comply with the Apple specification".

Supported formats and color spaces

MediaConvert ingests and outputs video in the following HDR formats:

- HDR10 (rec. 2020 color space)
- HLG (rec. 2020 color space)

MediaConvert ingests and outputs video in the following standard formats:

- SDR (rec. 601 color space)
- SDR (rec. 709 color space)

Supported color space conversions

Your input color space is set by your input video or by the values that you set for **Color space** and **Color space usage** in your input settings. For more information about the input color space settings, see Replacing inaccurate or missing HDR metadata.

For information about how to convert the color space, see Converting the color space.

MediaConvert supports the following color space conversions:

- From any supported HDR format to any other supported HDR format
- From any supported SDR color space to any other supported SDR color space
- From any supported SDR color space to any supported HDR format

> **Note**
>
> Converting from SDR to HDR doesn't upgrade the dynamic range of the video content itself. Therefore, the output is formatted as HDR but looks the same as it would if you created it as an SDR output.

- From any supported HDR format to any supported SDR color space

> **Note**
>
> When professional color graders convert an asset from HDR to SDR, they make artistic decisions about where to map colors from the larger space that don't exist in the smaller space. There is no standard formula to map these values automatically. The tone mapping technology that MediaConvert uses to do automatic conversion from HDR to SDR approximates the outcome of manually regrading from HDR to SDR. This automatic conversion works well with most content, but we recommend that you review your outputs to confirm the tone mapping results.

Creating HDR HLS outputs that comply with the Apple specification

For information about which Apple devices play back HDR content, see Find and watch movies with 4K, HDR, Dolby Vision, or Dolby Atmos in the Apple support documentation.

To create HDR outputs that comply with the Apple specification, you must make specific choices for your encoding settings. In the **Encoding settings** section for your output, specify these video settings as follows:

- **Video codec** – Choose **HEVC (H.265)**.
- **MP4 packaging type** – **HVC1**.
- **Profile** – Choose **Main10/High**.
- **Level** – Choose **5**.

> **Tip**
>
> The simplest way to find specific encoding settings in the console is to use your web browser's search on page function. For many browsers, this search is case sensitive.

Limitations for creating HDR10+ outputs

For information about which devices play back HDR 10+ content and test content, see https://hdr10plus.org.

You must use the following rules to create HDR 10+ outputs in AWS Elemental MediaConvert:

- Your input source video pixels must be HDR10, that is, **Color space** must be set to **Follow** or the Color corrector preprocessor must have **Color space conversion** set to **Force HDR 10**.
- Set **Video codec** to **HEVC (H.265)**.
- Set **Profile** to **Main10/Main** or **Main10/High**
- Set **Quality tuning level** to **Multi-pass HQ**.
- You cannot use the Dolby Vision preprocessor.

Passing through HDR content

By default, AWS Elemental MediaConvert sets your color space to **Follow**, which means that your output color space is the same as your input color space, even if the color space changes over the course of the video. Also by default, MediaConvert sets the output setting **Color metadata** to **Insert**, so that any color metadata is included in the output. If you want your output HDR to be the same as your input video, keep this setting and make sure that you choose HEVC for your codec and a 10-bit profile.

To pass through HDR content

1. Set up your transcoding job as usual. For more information, see the section called "Configuring jobs".
2. Make sure that the input **Color space** is set to the default value **Follow**.
   a. On the **Create job** page, in the **Job** pane on the left, choose **Input 1**.
   b. In the **Video selector** section on the right, for **Color space**, choose **Follow**.

3. For each HDR output, choose an appropriate codec and profile and make sure that **Color metadata** is set to the default value **Insert**.

    a. On the **Create job** page, in the **Job** pane on the left, choose the output, such as **Output 1**.

    b. In the **Encoding settings** section on the right, specify these video settings as follows:

    > **Tip**
    >
    > The simplest way to find specific encoding settings in the console is to use your web browser's search on page function. For many browsers, this search is case sensitive.

    - **Video codec** – Choose **HEVC (H.265)**.
    - **Profile** – Choose one of the 10-bit profiles: **Main10/Main**, **Main10/High**, **Main 4:2:2 10-bit/Main**, or **Main 4:2:2 10-bit/High**.
    - **Color metadata** – Choose **Insert**.

Replacing inaccurate or missing HDR metadata

If your input video is missing HDR metadata, or has HDR metadata that is wrong, you can add it or overwrite it in the input video settings. For HLG and HDR 10, you can specify the correct color space. For HDR 10, you can also specify accurate master display information.

> **Note**
>
> You use the input settings to supply metadata that is wrong or missing from your input files. Use output settings to do color space conversion.

To replace inaccurate or missing HDR metadata

1. Set up your transcoding job as usual. For more information, see the section called "Configuring jobs".
2. On the **Create job** page, in the **Job** pane on the left, choose your input.
3. In the **Video selector** section on the right, for **Color space**, choose the accurate color space for your input video.

4. For **Color space usage**, choose how AWS Elemental MediaConvert handles precedence between the input metadata and the value that you specify for **Color space**:

   - Choose **Force** if you want MediaConvert to use the color space that you specify for **Color space** regardless of whether it is specified in the input video metadata.
   - Choose **Fallback** if you want MediaConvert to use the color space that you specify for **Color space** only when the color space isn't present in the input video metadata.

     This option is useful when you reuse job settings with different input files, for example, when you use output presets or when you duplicate a job.

5. If your input is HDR 10, specify values for **HDR master display information** settings. MediaConvert displays these settings when you set **Color space** to **HDR 10**.

   These settings represent HDR 10 static metadata as specified in the standard SMPTE ST 2086 Mastering Display Color Volume. MediaConvert includes the values that you specify here in the metadata of your HDR 10 outputs.

   > **Note**
   >
   > Get your values for **HDR master display information** from a color grader. Appropriate values for these settings depend on the video content and are different for each input.

Converting the color space

If you want your output video to use a different color space than your input video, use color space conversion. Set up color space conversion in the output **Color corrector** settings.

MediaConvert supports the following input color spaces: Rec. 601, Rec. 709, HDR10, HLG 2020, P3DCI, and P3D65.

To convert the color space

1. Confirm that MediaConvert supports the conversion that you want to do. See the section called “Supported color space conversions”.
2. Set up your transcoding job as usual. For more information, see the section called “Configuring jobs”.
3. On the **Create job** page, in the **Job** pane on the left, choose your HDR output.

4. At the bottom of the **Encoding settings** section on the right, choose **Preprocessors**.
5. Choose **Color corrector** to display the color correction settings.
6. For **Color space conversion**, choose the color space that you want for your output.
7. If you are converting to HDR 10, specify values for the **HDR master display information** settings.

   These values don't affect the pixel values that are encoded in the video stream. They are intended to help the downstream video player display content in a way that reflects the intentions of the content creator.

Using image inserter in AWS Elemental MediaConvert

With the image inserter feature, you can insert a still image or motion image at a specified time. You can also display it as an overlay on the underlying video for a specified duration. This feature includes fade-in and fade-out capability and adjustable opacity.

You can set up an output with multiple overlays. Each overlay is independent of the others. For example, you can place a motion image logo in the video frame for the duration of the video, and include a still image HDR indicator only for portions of the file that are HDR. Each overlay has its own settings for opacity, fade-in and fade-out times, position on the frame, and length of time on the video. You can set up overlays to appear on the underlying video at the same time, or to overlap each other.

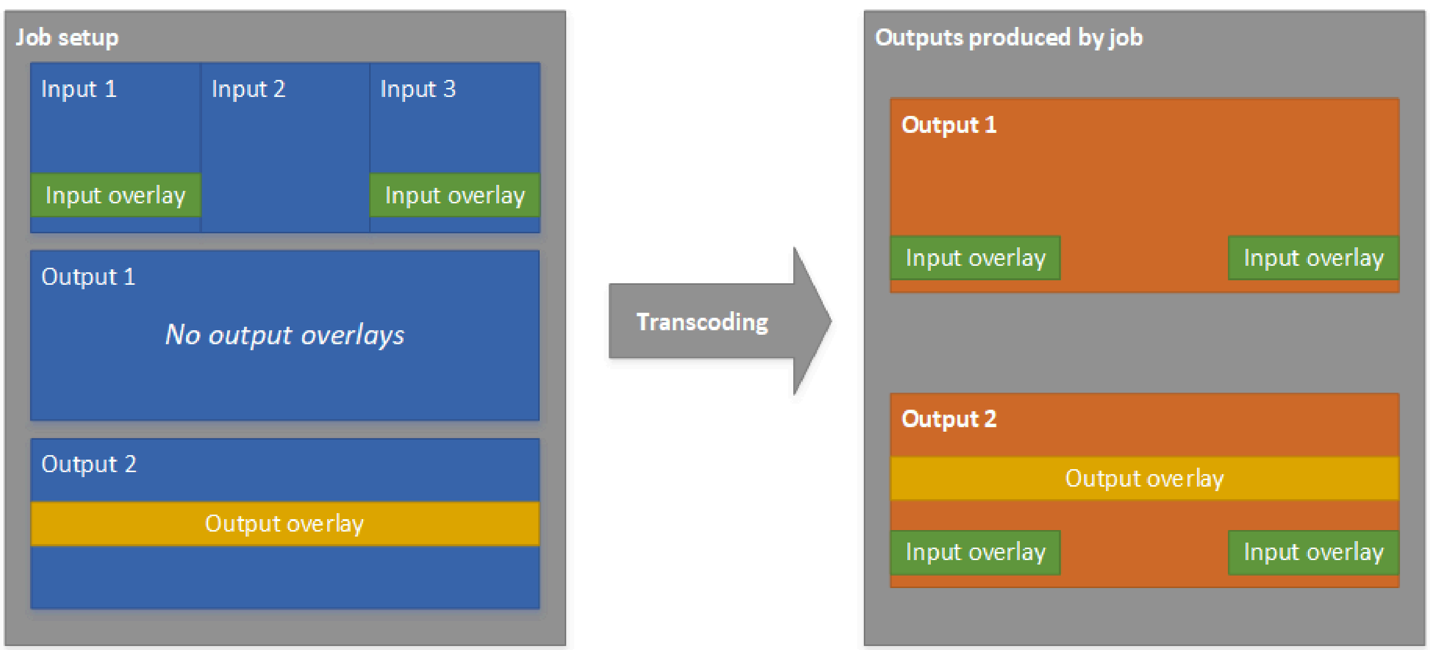
Topics

Choosing between input overlay and output overlay

You can add still image overlays to your inputs, your outputs, or both. Where you specify your image overlays affects where in your transcoded assets the overlays appear.

The following diagram shows how input and output overlays appear in the video files that are created by a job. Input overlays appear on all outputs, but only in the portions of the outputs that come from the input that has the overlay. Output overlays appear throughout an entire output, but only on the outputs that have the overlay.

> **Note**
>
> In this diagram, all overlays are specified for the entire duration of the input or output. You can instead specify a shorter overlay duration within that time.



Input overlays

Choose input overlay for the following situations:

- You want the same overlays on every output.
- You want an overlay on only the parts of your outputs that correspond to individual inputs.

These examples are situations where you would use input overlay:

- Some of your inputs already have your logo as an overlay and some of them don't. You want to add the logo only to the inputs that don't already have it.
- Some of your inputs are programming that you want your logo on. Other inputs are advertisements or blank slates that you don't want your overlay on.
- Your job has only one input. Your overlay should appear for the entire duration of the video and on every output of the job.

Output overlays

Choose output overlay for the following situations:

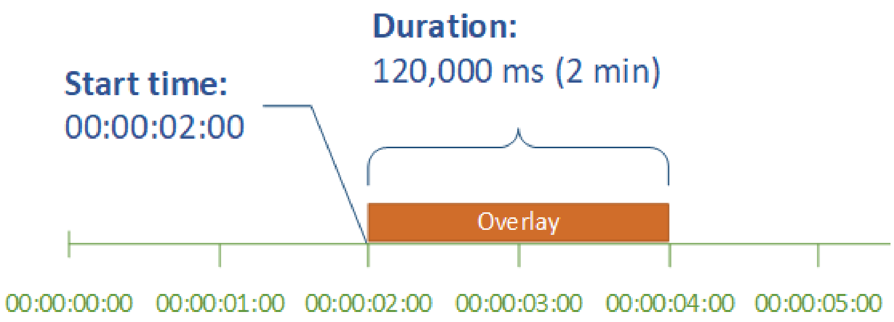
- You want overlays on some outputs but not others.
- You want different overlays on different outputs.
- You have multiple inputs, but you want the same overlay across all of them.

These examples are situations where you would use output overlay:

- You set up one of your outputs with high definition. You want to include an HD indicator in the corner of the frame on this output only.
- You are stitching together several films as separate inputs to create a single-asset film marathon. You want to put a image on all of them indicating that they are part of the larger marathon.

Placing your still image overlay

Whether your still image overlay is in input or output, set up the **Start time** and **Duration**. The following image shows how to specify settings so that an overlay starts two minutes into the video and remains on the video for two minutes. In the default settings, the overlay begins at the first frame of the input or output. The overlay remains on the video for the duration of the input or output.



Start time

Provide the timecode for the first frame that you want to have the overlay appear on. If you set up your overlay to fade in, the fade-in begins at the start time.

When you provide your start time, choose one of the following timelines to suit the overlay that you're using:

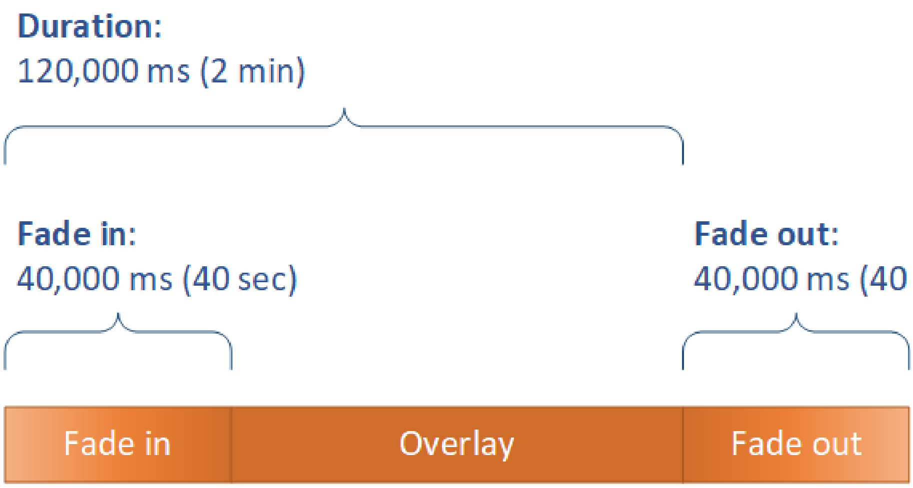
- For input overlays, **Start time** is relative to the input timeline. This timeline is affected by the input **Timecode source** setting.
- For output overlays, **Start time** is relative to the output timeline. This timeline is affected by the job-wide **Timecode configuration**, **Source** setting.

For more information about the input and output timelines, and the timecode settings that affect them, see the section called “How MediaConvert uses timelines to assemble jobs”.

> **Tip**
>
> For simplest setup, specify **Start time** counting from 00:00:00:00 as the first frame, and set both of the following settings to **Start at 0**:
>
> - **Timecode configuration**, **Source**, under the job-wide settings.
> - **Timecode source**, in the **Video selector** settings for each input.

Duration

Specify the length of time, in milliseconds, for the overlay duration to remain. This duration includes fade-in time, but not fade-out time, as the following image shows.



Requirements for the overlay file

Set up the image files that you want to insert over your video as follows:

- **File type**: Use .png or .tga.
- **Aspect ratio**: Use any aspect ratio; it doesn't need to match the aspect ratio of the underlying video.
- **Size in pixels**: Use any size. If the overlaid image is larger than the output video frame, the service crops the image at the edge of the frame.

> **Note**
>
> In jobs that scale the video resolution, whether your overlay scales with your video depends on where you specify the image overlay. For more information, see Sizing overlay for scaling.

Setting up still image overlays in outputs

Because you're setting up an output overlay, set up image insertion in each output where you want the service to overlay images on your video. For information about setting up an overlay that appears on all outputs, or on portions that correspond to only one input, see Choosing between input overlay and output overlay.

If you don't specify overlay start time and duration, the service puts the overlay on the entire output.

To set up a still image overlay in an output

1. Open the AWS Elemental MediaConvert console at https://console.aws.amazon.com/mediaconvert.
2. Set up your output groups and outputs for video and audio, as described in Configuring jobs in MediaConvert and Creating outputs.
3. For each output that you want to have a image overlay, do the following:

   a. On the **Create job** page, in the **Job** pane on the left, under **Output groups**, choose the appropriate output.

   b. Under **Encoding** settings, under the **Video** tab, find the **Preprocessors** section.

   c. Choose **Image inserter**. This displays an **Add image** button.

   d. For each image overlay that you want to include in the output, choose **Add image**, and then specify the overlay settings.

      For **Image location**, specify an input file that is stored in Amazon S3 or on an HTTP(S) server. For Amazon S3 inputs, you can specify the URI directly or choose **Browse** to select from your Amazon S3 buckets. For HTTP(S) inputs, provide the URL to your input file. For more information, see HTTP input requirements.

For details about the more complex output image overlay settings, see the following topics:

Sizing overlay for scaling

About specifying **Layer**

Setting up still image overlays in inputs

Because you are setting up an input overlay, set up image insertion in each input where you want the service to overlay imagess on your video. The overlays that you specify appear in every output. For information about setting up an overlay that appears on only specific outputs, see Choosing between input overlay and output overlay.

When don't specify overlay start time and duration, the service puts the overlay on the entire part of the output that corresponds to the input.

To set up a still image overlay in an output

1. Open the AWS Elemental MediaConvert console at https://console.aws.amazon.com/mediaconvert.
2. Specify your input files, as described in Configuring jobs in MediaConvert.
3. For each input that you want to have a image overlay, do the following:

   a. On the **Create job** page, in the **Job** pane on the left, under **Inputs**, choose the appropriate input.

   b. In the **Image inserter** section to the right of the **Job** pane, choose **Add image**, and then specify the overlay settings.

      For **Image location**, specify an input file that is stored in Amazon S3 or on an HTTP(S) server. For Amazon S3 inputs, you can specify the URI directly or choose **Browse** to select from your Amazon S3 buckets. For HTTP(S) inputs, provide the URL to your input file. For more information, see HTTP input requirements.

For details about the more complex input image overlay settings, see the following topics:

Sizing overlay for scaling

About specifying **Layer**

Sizing your overlay for scaling

In jobs that scale the video resolution, whether your overlay scales with your video depends on where you specify the image overlay. Motion image overlays and input overlays scale with the video; output overlays don't.

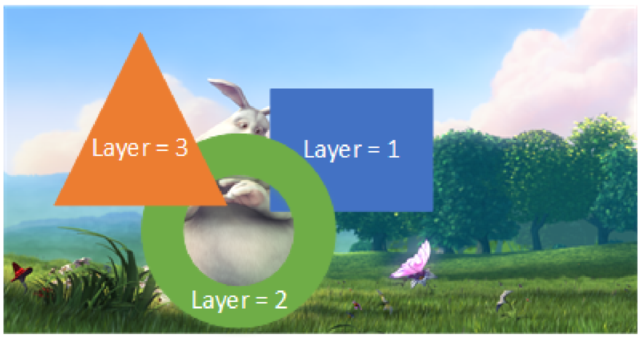
For example, suppose that the input video for your job is 1080 x 1920 and you specify three outputs at 720 x 1280, 480 x 640, and 360 x 480. Your square logo would be 10% of the width of your frames, and your overlay images would have the following resolutions:

- For a motion image overlay or an input image overlay, provide an image that is 108 x 108. The service appropriately sizes each overlay on each output.
- For an output image overlay on your 720 x 1280 output, provide an image that is 72 x 72.
- For an output image overlay on your 480 x 640 output, provide an image that is 48 x 48.
- For an output image overlay on your 360 x 480 output, provide an image that is 36 x 36.

About specifying the overlay Layer

The **Layer** setting specifies how overlapping image overlays appear in the video. The service overlays images with higher values for **Layer** on top of overlays with lower values for **Layer**. Each overlay must have a unique value for **Layer**; you can't assign the same layer number to more than one overlay.

The following illustration shows how the value for **Layer** affects how a image overlay appears in relation to other overlays. The triangle has the highest value for **Layer** and appears on top, obscuring the video frame and all image overlays with lower values of **Layer**.



To specify a value for the Layer setting

1. Set up your image overlay as described in Image insertion.
2. For **Layer**, enter a whole number from 0 to 99.

> **Note**
>
> You can use each number only once. Each image overlay must have its own layer.

Using Kantar for audio watermarking in AWS Elemental MediaConvert outputs

Kantar is a company that provides and tracks unique identifiers for media content in the form of audio watermarks that are audible to machines but not to humans. After you establish a relationship with Kantar, you can use AWS Elemental MediaConvert to encode these watermarks into your content, for audience measurement and anti-piracy applications.

To use MediaConvert to encode Kantar watermarks, you use AWS Secrets Manager to store your Kantar credentials. These credentials are the username and password that you set up with Kantar. Then you grant the MediaConvert service permission to read those credentials, so that MediaConvert can communicate with the Kantar servers to request your watermarks and to provide log information. In your MediaConvert job, you provide your Kantar license information and information about the media asset that you're encoding. For more information about these steps, see the following topics.

Topics

- Getting a Kantar watermarking license
- Storing your Kantar credentials in AWS Secrets Manager
- Granting IAM permissions to your Kantar credentials
- Setting up your MediaConvert job for Kantar watermarking

Getting a Kantar watermarking license

To use Kantar audio watermarking with MediaConvert, you must first establish a relationship with Kantar and acquire a license from them. Kantar provides the following guidance for doing so.

In order to get a watermarking embedding license, contact Kantar Media support with following information:

- Product name and version
- Customer name
- Country
- If different, country of broadcast
- Channel(s) to be watermarked
- Customer internal name for the hardware platform
- AuthorisationCode for each piece od hardware, or login contact for the online solution

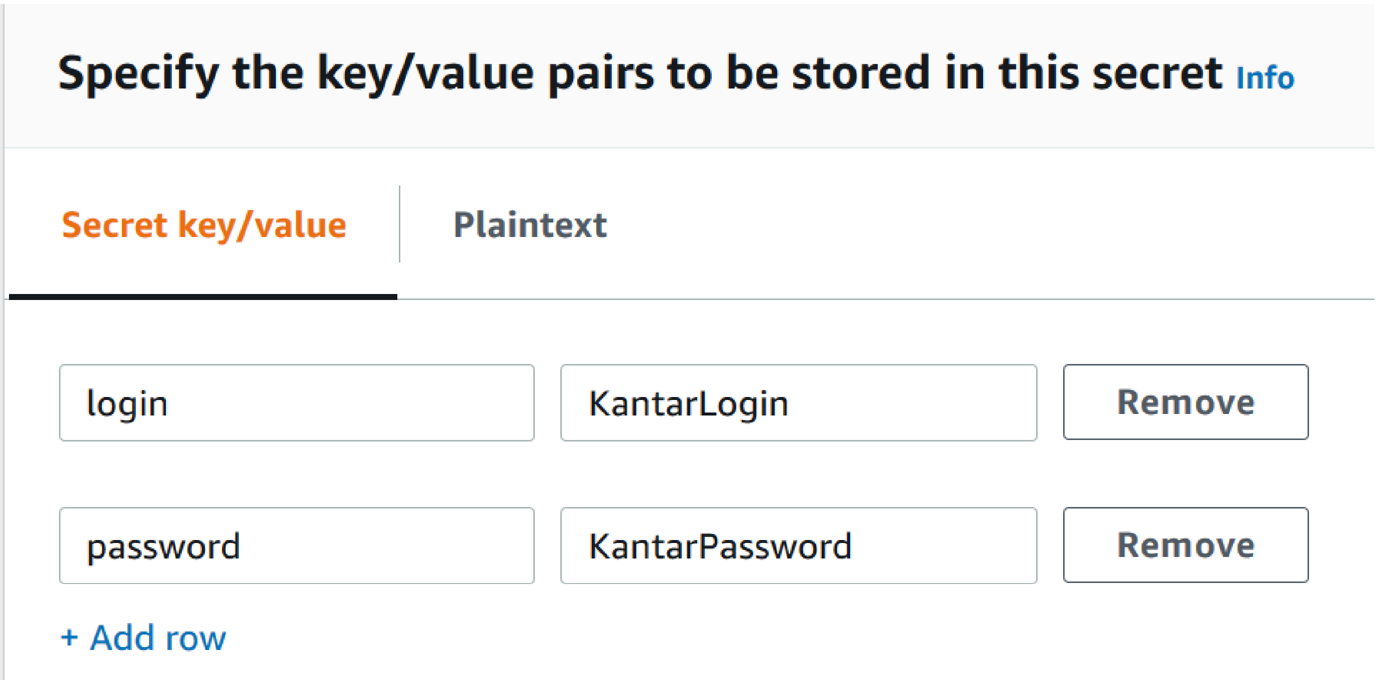
Storing your Kantar credentials in AWS Secrets Manager

After you establish a relationship with Kantar, store your Kantar username and password in an AWS Secrets Manager secret. You can find a step-by-step tutorial in the topic Create and store your secret in AWS Secrets Manager in the *AWS Secrets Manager User Guide.*

To store your Kantar credentials in a Secrets Manager secret

Follow the tutorial linked in the introduction to this procedure, with the following differences.

1. Make sure that you use the same AWS Region for Secrets Manager that you use for your MediaConvert job.
2. In the **Specify key/value pairs to be stored in the secret** section, set up two key-value pairs. One holds your Kantar login name and the other holds your password.

   a. Type **`login`** in the left field and then type your Kantar login name in the right field.
   b. Choose **Add row** to bring up a second pair of entry fields.
   c. Type **`password`** in the left field and then type your Kantar password in the right field.

   The plaintext version of your secret should look like this: `{ "login": "KantarLogin", "password": "KantarPassword" }`, where KantarLogin is the username for your Kantar credentials and KantarPassword is the password for your Kantar credentials.
3. In the **Secret name and description** section, for **Secret name**, use a name that you will associate with Kantar, such as **`KantarCreds`**.

> **Note**
>
> You provide the name of this secret in your MediaConvert job settings.

4. For **Description**, you might want to note that this is the secret that you use with MediaConvert for encoding your Kantar watermarks.
5. After you store the secret, copy the ARN of your secret to use when you grant IAM permissions to MediaConvert to access the secret.

   a. On the Secrets Manager **Secrets** page, choose the name of your secret.

   b. Copy and save the string you see listed under **Secret ARN**.

   > **Note**
   >
   > You use this ARN when you grant MediaConvert permissions to get this secret.

Granting IAM permissions to your Kantar credentials

When you use AWS Elemental MediaConvert, you specify an IAM service role that grants permissions to the service to access the resources it needs to run your job. For example, your MediaConvert service role grants MediaConvert permissions to read your job input files from Amazon S3. For information about setting up that service role, see Setting up IAM permissions .

To encode Kantar watermarks, add permissions to this service role to grant MediaConvert access to read the AWS Secrets Manager secret that holds your Kantar credentials.

To grant MediaConvert permission to read your Kantar credentials

1. Create a policy that grants permission to read your Secrets Manager secret.

   a. Make sure that you have the ARN to the Secrets Manager secret that you created in the previous topic.

   b. Open the IAM console at https://console.aws.amazon.com/iam/.

   c. In the navigation pane on the left, under **Access management**, choose **Policies**.

   d. Choose **Create policy**.

   e. On the **Create policy** page, next to **Service**, choose **Choose a service**.

f. In the search field, type `secrets` and then choose **Secrets Manager** from the results.

g. In the **Filter actions** search field, type `GetSecretValue` and then choose **GetSecretValue** from the results.

h. In the **Resources** section, next to **Secret**, choose **Add ARN**.

i. On the **Add ARN(s)** page, next to **Specify ARN for Secret**, choose **List ARNs manually**.

j. In the **Type or paste a list of ARNs** section, paste the ARN for your Kantar credentials secret that you copied at the end of the procedure in the previous topic.

k. Choose **Add**.

l. At the bottom of the **Create policy** page, choose **Next: Tags**.

m. Choose **Next: Review**.

n. Under **Review policy**, for **Name** type a name that will help you remember the purpose of this policy, such as `GetKantarCreds`.

o. Optionally, for **Description**, jot a note to yourself for later. For example, you might write "This provides MediaConvert permission to read my Kantar credentials."

p. Choose **Create policy**.

2. Attach the policy to your MediaConvert role.

a. In the navigation pane on the left, under **Access management**, choose **Roles**.

b. From the list of roles, choose the name of the role that you use with your MediaConvert job. This role is often **MediaConvert_Default_Role**.

c. On the role **Summary** page, on the **Permissions** tab, choose **Attach policies**.

d. In the search field, type the name of the policy you created, such as `GetKantarCreds`.

e. In the results list, choose the check box next to the policy name.

f. Choose **Attach policy**.

g. On the **Summary** page for the role, review the list of policies and confirm that your policy that grants permission to get your Kantar credentials appears there.

Setting up your MediaConvert job for Kantar watermarking

After you've established your relationship with Kantar, stored your Kantar credentials in AWS Secrets Manager, and granted permission to AWS Elemental MediaConvert to get these credentials, set up your MediaConvert job to encode the Kantar watermarks.

Feature restrictions

When you set up your job, note these restrictions on your other settings.

- You can't use Kantar watermarking and Nielsen watermarking in the same job.
- Your output audio sample rate must be 48 kHz or greater.

To set up your job to encode Kantar watermarks

1. Set up your job as usual. For more information, see Configuring jobs in MediaConvert.
2. On the **Create job** page, in the **Job** pane on the left, under **Job settings**, choose **Partner integrations**.
3. Enable **Kantar SNAP file watermarking**.
4. Provide values for the Kantar settings.

    a. For **Credentials secret name**, type the name of the Secrets Manager secret that you created to store your Kantar credentials. For example, **`KantarCreds`**.

    b. For **Kantar license ID**, type the license ID that Kantar provides you.

    c. For **Channel name**, type one of the channel names that are listed in your Kantar audio license.

    d. For **Content reference**, type the unique identifier that Kantar uses for the asset that you're encoding.
5. Confirm that the service role you've specified in the job is the same one that you, in the previous topic, attached permissions to that grant access to your Kantar credentials. If that role Is MediaConvert_Default_Role, you don't need to choose it explicitly, because MediaConvert will use that role by default. To specify the role if it has a different name, do the following:

    a. In the **Job** pane on the left, choose **AWS integration**.

    b. In the **Service access** section, find **Service role**. Confirm that the specified role is the one with the correct permissions.

Motion image inserter in AWS Elemental MediaConvert

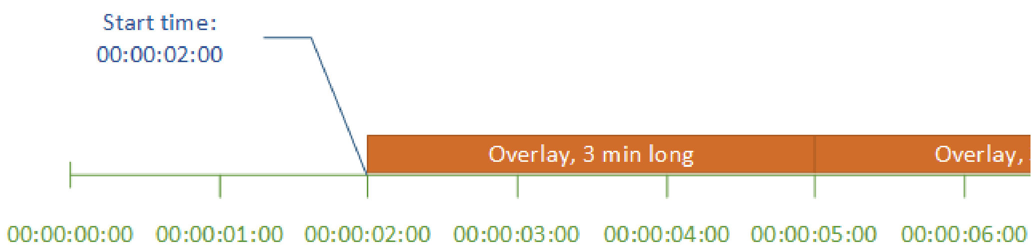
In this guide, you'll learn how to set up motion graphic overlays with AWS Elemental MediaConvert. Motion graphic overlays appear in all outputs.

In the default setting for **Start time**, the overlay starts at the first frame of each output. In the default setting for **Playback**, the overlay remains on the video for the duration of the motion graphic (played once). However, you can change the default setting by specifying an overlay start time, or by setting playback to repeat. This guide shows you how.

Specify motion graphic overlay start time and playback

You can specify motion graphic overlay **Start time** and **Playback** settings instead of using the default setting. The following information shows how to specify overlay start time for a video and to repeat it continuously (loop).

In the following image, the motion graphic overlay setting is three minutes long. The motion graphic playback is set to repeat until the end of the output.



Start time settings for motion overlays

Provide the timecode for the first frame where you want the motion overlay to appear. This timecode is relative to your input timeline.

Storage management

For input overlays, **Start time** is relative to the input timeline. This timeline is affected by the input **Timecode source** setting.

For more information about the input and output timelines, and the timecode settings that affect them, see the section called "How MediaConvert uses timelines to assemble jobs". For jobs with multiple inputs, MediaConvert places the motion overlay on each input, according to the input timeline for that input. After you specify **Start time** once, MediaConvert applies that value to all inputs.

> **Tip**
>
> To simplify setup, specify **Start time** counting from 00:00:00:00 as the first frame, and set both of the following settings to **Start at 0**:
>
> - **Timecode configuration**, **Source**, under the job-wide settings.
> - **Timecode source**, in the **Video selector** settings for each input.

Playback settings for motion graphic overlays

For playback settings on motion graphic overlays, you have two options. You can set your overlay to play once through the duration of the motion graphic or to loop from the start time to the end of the output. The duration of a .mov motion graphic is built into the .mov file, which has a set number of frames and a defined frame rate.

When a motion graphic is a set of .png images, determine the duration of the overlay by how many images you provide and the frame rate that you specify. The duration in seconds is the number of frames divided by the frame rate, in frames per second. For example, if your frame rate is 30 fps and you provide 600 images, the duration of the motion overlay is 20 seconds.

For jobs with multiple inputs, MediaConvert places the motion overlay on each input at the time that you specify for **Start time**. Depending on what you choose for **Playback**, MediaConvert either plays the overlay once or until the end of the input. When you specify **Playback** once, MediaConvert applies that value to all inputs.

Requirements to set up motion graphic overlay files

The following table describes how to set up motion graphic overlay files.

Motion graphic file requirement	Description
File type	QuickTime (.mov) • Container: QuickTime • Codec: QuickTime Animation (RLE) • Color space: RGBA

Motion graphic file requirement	Description
	Sequential PNG (.png) • Make sure that the names of the .png files end with sequential numbers that specify the order that they are played in. For example, overlay_000.png, overlay_001.png, overlay_002.png, and so on. • Pad your initial file name with enough zeros to complete the sequence. For example, if the first image is overlay_0.png, there can be only 10 images in the sequence, with the last image being overlay_9.png. However, if the first image is overlay_00.png, there can be 100 images in the sequence. • Make sure that the number of images in your series matches the frame rate and your intended overlay duration. For example, if you want a 30-second overlay at 30 fps, you should have 900 .png images. • Requires an alpha channel.
Frame rate	QuickTime (.mov) • Use any frame rate. The frame rate that you use doesn't have to match the frame rate of the underlying video. Sequential PNG (.png) • Use any frame rate. The frame rate that you use doesn't have to match the frame rate of the underlying video. • Specify the frame rate when you set up the overlay.

Motion graphic file requirement	Description
Aspect ratio	Use any aspect ratio. It doesn't have to match the aspect ratio of the underlying video.
Size in pixels	Use any size. MediaConvert scales the motion graphic with any outputs that have video scaling.

Setting up motion graphic overlays

Motion graphic overlays apply to every output in the job. Therefore, set them up as a processor in the settings that apply to the entire job.

You can set up still graphic overlays that appear only on individual outputs. For information, see Choosing between input overlay and output overlay.

To set up a motion graphic overlay

1. Open the AWS Elemental MediaConvert console at https://console.aws.amazon.com/mediaconvert.
2. Set up your job, as described in Configuring jobs in MediaConvert.
3. On the **Create job** page, in the **Job** pane on the left, under **Job settings**, choose **Settings**.
4. In the **Global processors** section to the right of the **Job** pane, enable **Motion image inserter**.
5. For **Input**, specify your motion graphic file name. If you're using a series of .png files, provide the file name of the first image.
6. Specify values for the other fields. For more information about these fields, choose the **Info** link on the console next to **Motion image inserter**.

Working with Nielsen to do audio watermarking in AWS Elemental MediaConvert outputs

Nielsen is a company that tracks how often video assets are watched by viewers. One form of that tracking uses tones in a media asset's audio that are audible to machines but not humans. These tones are encoded directly in the audio stream and can also be cued in the metadata. To use this Nielsen audio watermarking with MediaConvert, you must first establish a relationship with Nielsen.

MediaConvert supports Nielsen audio watermarking in these ways:

- *PCM to ID3 watermarking*: With PCM to ID3 watermarking, MediaConvert translates watermarking that already exists in your input audio stream to markers in your output's ID3 metadata.
- *Non-linear watermarking*: With non-linear watermarking, MediaConvert inserts integers, called *TICs*, into the PCM audio stream of the asset. At the locations of these TICs, MediaConvert also encodes audio tones.

  With non-linear watermarking, your input must start without watermarking. AWS Elemental MediaConvert receives the TICs from a Nielsen SID/TIC server that you set up in the AWS Cloud.

> **Note**
>
> If you want both types of watermarking, you must run your job twice. First create an output with non-linear watermarking, and then use that output as input to the next job to add ID3 watermarking from your PCM stream. You can't enable both kinds of watermarking in a single job.

Topics

- Setting up your MediaConvert job for PCM to ID3 metadata
- Setting up your MediaConvert job for non-linear watermarking
- How AWS Elemental MediaConvert interacts with your Nielsen SID/TIC server in the AWS Cloud

Setting up your MediaConvert job for PCM to ID3 metadata

To use this feature, your input must have PCM audio that contains Nielsen watermarks. You provide your Nielsen distributor ID to MediaConvert and then, during the transcode, MediaConvert inserts the watermark information into the ID3 metadata of your output.

Feature restriction

You can put Nielsen watermarking information into the ID3 metadata of outputs in the following output groups only:

- Apple HLS
- File group, when your output container is MPEG-2 Transport Stream

To enable PCM to ID3 metadata (console)

1. Set up your job inputs and outputs as described in Configuring jobs in MediaConvert.
2. Enable PCM to ID3 metadata in the job-wide settings.
   a. On the **Create job** page, in the **Job** pane on the left, under **Job settings** choose **Settings**.
   b. In the **Partner integrations** section on the right, choose **Nielsen PCM to ID3 metadata**.
   c. For **Distributor ID**, provide the ID that Nielsen assigned to your organization.
3. Enable PCM to ID3 metadata in the outputs where you want it. Do these steps for each **Apple HLS** output that you want to have ID3 metadata.
   a. In the **Job** pane on the left, choose the output.
   b. In the **Output settings** section on the right, expand the section **Transport stream settings**.
   c. For **Nielsen ID3**, choose **Insert**.
4. Do these steps for each **File group** output that you want to have ID3 metadata.
   a. In the **Job** pane on the left, choose the output.
   b. In the **Output settings** section on the right, expand the section **Container settings**.
   c. Scroll to the section **PID controls**.
   d. For **Nielsen ID3**, choose **Insert**.

To enable PCM to ID3 metadata (API, CLI, and SDK)

If you use the API, CLI, or an SDK, specify the relevant settings in your JSON job specification and then submit it programmatically with your job. For more information about submitting your job programmatically, see one of the introductory topics of the *AWS Elemental MediaConvert API Reference*:

- Getting started with AWS Elemental MediaConvert using the AWS SDKs or the AWS CLI

- Getting started with AWS Elemental MediaConvert using the API

- Use the MediaConvert console to generate your JSON job specification. We recommend this approach, because the console functions as an interactive validator against the MediaConvert job schema. Follow these steps to generate your JSON job specification using the console:

   a. Follow the previous procedure for the console.

   b. In the **Job** pane on the left, under **Job settings**, choose **Show job JSON**.

Find additional information, including where each setting belongs in the job settings structure, in the *AWS Elemental MediaConvert API Reference*. Links in this list go to information about the setting in that document:

- **Nielsen PCM to ID3 metadata** (`nielsenConfiguration`)
- **Distributor ID** (`distributorId`)
- **Nielsen ID3**, for outputs in an **Apple HLS** output group (`nielsenId3`, child of `m3u8Settings`)
- **Nielsen ID3**, for outputs in a **File group** output group (`nielsenId3`, child of `m2tsSettings`)

Setting up your MediaConvert job for non-linear watermarking

To use this feature, you must first establish a relationship with Nielsen and set up a Nielsen SID/TIC server in the AWS Cloud. Contact Nielsen to download their SID/TIC server software, generate a WRR license file, and receive installation and setup instructions. For an overview of how the infrastructure works, see How AWS Elemental MediaConvert interacts with your Nielsen SID/TIC server in the AWS Cloud.

To set up Nielsen non-linear watermarking (console)

1. Set up a Nielsen SID/TIC server system in the AWS Cloud. For more information, contact Nielsen.
2. Set up an Amazon S3 bucket to hold your Nielsen metadata .zip file. MediaConvert writes the metadata to this bucket.
3. Set up your job inputs and outputs as described in Configuring jobs in MediaConvert.
4. On the **Create job** page, in the **Job** pane on the left, under **Job settings** choose **Settings**.
5. In the **Partner integrations** section on the right, choose **Nielsen non-linear watermarking**.

6. Provide values for the settings that become visible when you enable **Nielsen non-linear watermarking**. For instructions and guidance about each of these settings, choose the **Info** link next to the setting label.
7. Choose **Create**, at the bottom of the page, to run your job.
8. Transfer the data in your metadata Amazon S3 bucket to Nielsen, according to their instructions.

To set up Nielsen non-linear watermarking (API, CLI, and SDK)

If you use the API, CLI, or an SDK, specify the relevant settings in your JSON job specification and then submit it programmatically with your job. For more information about submitting your job programmatically, see one of the introductory topics of the *AWS Elemental MediaConvert API Reference*:

- Getting started with AWS Elemental MediaConvert using the AWS SDKs or the AWS CLI
- Getting started with AWS Elemental MediaConvert using the API

- Use the MediaConvert console to generate your JSON job specification. We recommend this approach, because the console functions as an interactive validator against the MediaConvert job schema. Follow these steps to generate your JSON job specification using the console:

  a. Follow the previous procedure for the console.
  b. In the **Job** pane on the left, under **Job settings**, choose **Show job JSON**.

Find additional information, including where each setting belongs in the job settings structure, in the *AWS Elemental MediaConvert API Reference*. Links in this list go to information about the setting in that document:

- **Nielsen non-linear watermarking** (`nielsenNonLinearWatermark`)
- **Source watermark status** (`sourceWatermarkStatus`)
- **Watermark types** (`activeWatermarkProcess`)
- **SID** (`sourceId`)
- **CSID** (`cbetSourceId`)
- **Asset ID** (`assetId`)
- **Asset name** (`assetName`)

- **Episode ID** (`episodeId`)
- **TIC server REST endpoint** (`ticServerUrl`)
- **ADI file** (`adiFilename`)
- **Metadata destination** (`metadataDestination`)
- **Share TICs across tracks** (`uniqueTicPerAudioTrack`)

How AWS Elemental MediaConvert interacts with your Nielsen SID/TIC server in the AWS Cloud

The following diagram shows, at a high level, how MediaConvert interacts with the other parts of the system.

There are four major parts of the system that you set up by following instructions from Nielsen:

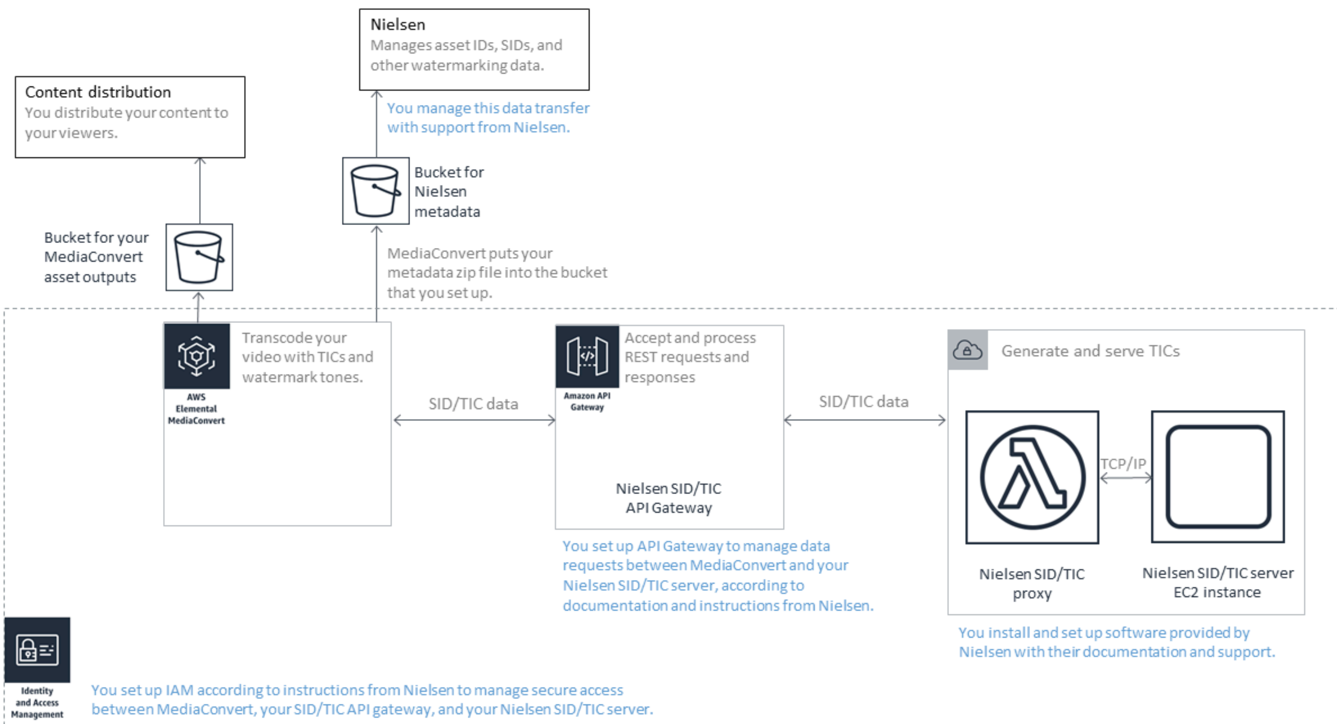
- Transfer your Nielsen metadata .zip file. MediaConvert writes this file to the Amazon S3 bucket that you create. You provide this bucket URL for the setting **Metadata destination** (`metadataDestination`) when you set up your job.

> **Note**
>
> This bucket is different from the one that you set up for the media asset outputs of your job. For information about that bucket, see Setting up, which is a sub-topic of the Getting started chapter of this guide.

- Set up an Amazon EC2 instance and an AWS Lambda proxy in an Amazon VPC. Then install the SID/TIC server software and license from Nielsen.
- Use Amazon API Gateway to set up a gateway to manage requests and responses between MediaConvert and your Nielsen SID/TIC server.

  This should result in a REST endpoint that you provide for the setting **TIC server REST endpoint** (`ticServerUrl`) when you set up your job.
- Use AWS Identity and Access Management (IAM) to manage access and authentication between MediaConvert, your API gateway, and your SID/TIC server.

> **Note**
>
> The roles and access you set up for this data sharing is different from the IAM permissions you set up to allow MediaConvert to access your input and output Amazon S3 buckets. For information about that setup, see Setting up IAM permissions , which is a sub-topic of the Getting started chapter of this guide.



Using the QVBR rate control mode

The rate control mode that you choose for your output determines whether the encoder uses more data for complex parts of your video or maintains a constant amount of data per frame. This chapter provides guidance for choosing the right rate control mode for your asset, depending on how you plan to distribute it. In general, you get the best video quality for a given file size by using quality-defined variable bitrate (QVBR) for your rate control mode.

Comparison of QVBR with other rate control modes

The rate control mode that you choose depends on the way that you will distribute your asset. AWS Elemental MediaConvert offers the following choices for bitrate mode:

Quality-Defined Variable Bitrate (QVBR) Mode

Choose this mode for distribution over the internet (OTT) and for video on-demand (VOD) download. For higher video quality for your file size, always choose this mode except in the following cases:

- You need your bitrate to be constant, for example, for distribution over fixed-bandwidth networks
- You need your total file size to not drop below the size that you specify, for example, to comply with contractual or regulatory requirements

When you choose QVBR, the encoder determines the right number of bits to use for each part of the video to maintain the video quality that you specify. You can keep the QVBR quality level blank to let the encoder determine the appropriate quality level based on the input video frames. You can use the same QVBR settings for all your assets; the encoder automatically adjusts the file size to suit the complexity of the video. For more information, see Guidelines for using QVBR.

Constant Bitrate (CBR) Mode

Choose CBR only if you need the asset's bitrate to remain constant over time. For example, you might need a constant bitrate if you distribute your assets over limited, fixed bandwidth networks.

When you choose CBR, the encoder caps the file size and quality with the value that you set for **Bitrate**. The encoder uses the same number of bits for all parts of the video.

Variable Bitrate Mode (VBR)

Choose VBR if you distribute your asset over a network that allows for a changing bitrate, like the internet, but you need to specify the total file size of your asset.

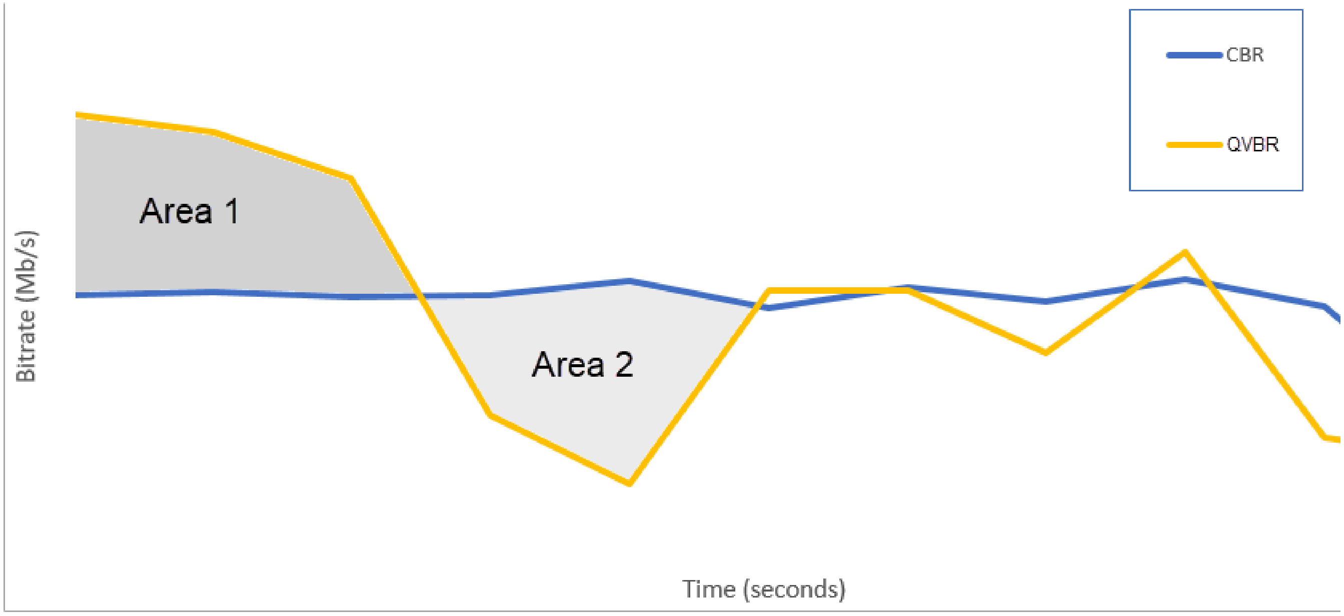
> **Note**
>
> With QVBR, if you set up your output for multi-pass encoding, you can optionally specify a maximum average bitrate that caps the total file size of your output. Only choose VBR if your file size can't be smaller than the size you specify.

With VBR, you specify the asset's average bitrate; the encoder allocates bits so that more bits go to complex parts of the video. The total file size (excluding container, packaging, and audio data) works out to the average bitrate that you specify (in bits per second) times the length of the asset (in seconds).

When you use VBR, you get best results if you adjust your average bitrate to suit the complexity of each asset.

The following graph illustrates how the varying bitrate modes (QVBR and VBR) save unnecessary bits and provide better quality compared to CBR. The graph shows QVBR versus CBR, but the same principle applies to VBR.

In the parts of the graph where the QVBR line is above the CBR line, as in the part labeled Area 1, the CBR capped bitrate limits video quality below that of other scenes, so QVBR gives you more consistent quality. In the parts where the QVBR line drops below the CBR line, as in the part labeled Area 2, a low bitrate is sufficient for the same video quality, so QVBR saves bits and provides the opportunity for cost savings in storage and distribution through your content delivery network (CDN).



Guidelines for using quality-defined variable bitrate mode

When you use QVBR, you can specify the quality level for your output and the maximum peak bitrate. For reasonable values of those settings, the encoder chooses how many bits to use for each part of the video. When you apply the same settings to several assets, your job outputs for simpler assets (such as cartoons) have smaller file sizes than your outputs for visually complex assets (such as high-motion sports with brightly dressed crowds in the background).

This section provides information about the QVBR settings. The following table provides a set of recommended values to get started with. Specify your values for these settings when you create your outputs, as described in Configuring jobs in MediaConvert. For more information about each setting, choose a topic from the list that follows the table.

Resolution	Width	Height	QVBR quality level	Max bitrate
1080p	1920	1080	9	6000000
720p	1280	720	8	4000000
720p	1280	720	7	2000000
480p	852	480	7	1000000

Resolution	Width	Height	QVBR quality level	Max bitrate
360p	640	360	7	700000
288p	512	288	7	350000

With all resolutions, don't specify a value for **Max average bitrate** unless you need to guarantee a total file size cap. When you specify a maximum average bitrate, it reduces the benefit that QVBR provides in video quality to file size ratio. To use **Max average bitrate**, you have to first set **Quality tuning level** to **Multi-pass HQ**.

If you aren't using **Max average bitrate**, and you don't need multi-pass encoding for other reasons, set **Quality tuning level** to **Single-pass HQ**.

> **Note**
>
> Multi-pass encoding is a professional tier feature. For more information about MediaConvert pricing tiers, see MediaConvert pricing.

Setting QVBR quality tuning level

Default QVBR quality level

You can keep the QVBR quality level blank to let the encoder automatically determine the appropriate quality level based on the input video frames. When you choose the default option, the encoder produces a more consistent quality across the entire video instead of a specified target quality level. If you choose this option, you can’t maintain a differentiated quality level based on the intended viewing device (for example, large-screen TV, PC or tablet, or smartphone). The encoder determines the appropriate quality level for you, based on the characteristics of your input video, to maintain a consistent video quality.

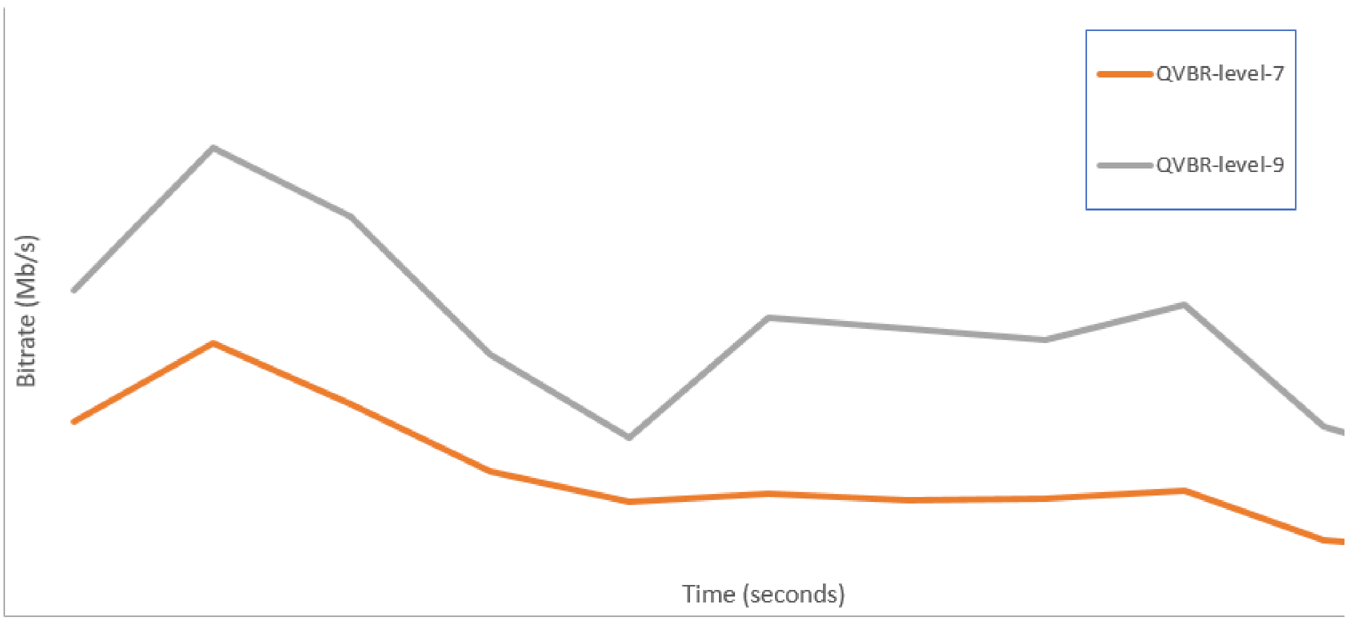
Custom QVBR quality level

You can specify the **QVBR quality level** on a scale between 1 and 10. The encoder determines the right number of bits to use for each part of the video to maintain the video quality that you specify.

The best value for an output depends on how the output will be viewed. In general, set **QVBR quality level** as shown in the following table.

Intended viewing device	Recommended QVBR quality level for 720p/1080p
Large-screen TV	8 or 9
PC or tablet	7
Smartphone	6

The following graph shows how changing the quality level affects the bitrate that the encoder uses for different parts of the video. While the lines for both level 7 and level 9 spike and drop in the same places, the encoder uses more bits total when the quality is set higher.



Including SCTE-35 markers in AWS Elemental MediaConvert outputs

SCTE-35 markers indicate where downstream systems can insert other content (usually advertisements or local programs). You can include SCTE-35 markers in transport stream (TS), DASH, HLS, and CMAF outputs.

AWS Elemental MediaConvert puts SCTE-35 markers into your outputs in one of two ways:

- The service passes markers through from your input to the output. For more information, see Passing through SCTE-35 markers from your input.
- The service inserts markers at the points that you specify in an Event Signaling and Management (ESAM) XML document. For more information, see Specifying SCTE-35 markers using ESAM XML.

Regardless of which way you put in the SCTE-35 markers, for outputs that have them, you can optionally do the following:

- You can have the service blank out audio and video during the ad avails indicated by the SCTE-35 markers. For more information, see Enabling ad avail blanking.
- For HLS outputs, you can have the service include SCTE-35 information in your output HLS manifest. For more information, see Including SCTE-35 information in your HLS manifest. MediaConvert doesn't write SCTE-35 information to DASH manifests.

> **Note**
>
> MediaConvert does not process information from input manifests.

By default, the service doesn't pass through SCTE-35 markers from your input. When you set up your job to pass through markers from the input or from an ESAM document, by default, the service doesn't include SCTE-35 information in HLS manifests or do ad avail blanking.

Feature Limitations

Limitations to SCTE-35 support are as follows:

- You can either specify insertion points using ESAM XML or pass through SCTE-35 messages from the input. You can't do both.
- AWS Elemental MediaConvert supports only time_signal messages, not splice_insert messages.
- The service inserts SCTE-35 messages only into the following outputs:
  - Outputs in **File group** output groups with **MPEG-2 Transport Stream** set for **Container**.

    Set the container for each output under **Output settings**, **Container**.
  - Outputs in **DASH ISO** output groups.

    - Outputs in **Apple HLS** output groups.
    - Outputs in **CMAF** output groups.
- The service forces Instantaneous Decoder Refresh (IDR) frames at the insertion points specified in your ESAM XML document for the following output codecs: MPEG-2, MPEG-4 AVC (H.264), or HEVC (H.265).

  In the MediaConvert console, go to **Encoding settings**, **Video**, and then set the codec for each output in the **Video codec** section.
- **DASH ISO** and **CMAF DASH** output groups only support single-period manifest outputs.

Topics

Passing through SCTE-35 markers from your input

You can include time_signal SCTE-35 markers from your input in any output that has a transport stream container. These outputs might be in an HLS package, or they might be standalone files wrapped in an MPEG2 transport stream (M2TS) container.

To pass through SCTE-35 markers from the input to an output (console)

1. Open the MediaConvert console at https://console.aws.amazon.com/mediaconvert.
2. Choose **Create job**.
3. Set up your input, output groups, and outputs for video and audio, as described in Configuring jobs in MediaConvert and Creating outputs.
4. Choose an output under either **File group**, **DASH ISO**, **Apple HLS**, or **CMAF**.
5. Under **Container settings** (for **File group**, **DASH ISO**, or **CMAF** outputs), or **Transport stream settings** (for **Apple HLS** outputs), find **SCTE-35 source**, and then choose **Passthrough**.
6. Optional. For outputs in a **File group** output group, when you set **Output settings**, **Container** to **MPEG-2 Transport Stream** (M2TS), you can enter a value for **SCTE-35 PID** that is different from the default **500**.

A PID, or packet identifier, is an identifier for a set of data in an MPEG-2 transport stream container. PIDs are used by downstream systems and players to locate specific information in the container.

7. Optional. For outputs in an **Apple HLS** output group, you can set up the job to include ad markers in the manifest. For more information, see Including SCTE-35 information in your HLS manifest.

Specifying SCTE-35 markers using ESAM XML

If your input video doesn't contain SCTE-35 markers, but you need to specify ad insertion points in your outputs, you can provide Event Signaling and Management (ESAM) XML documents in your AWS Elemental MediaConvert job settings. When you do, MediaConvert conditions your outputs with IDR (Instantaneous Decoder Refresh) frames at the insertion points that you specify in the document. In outputs that are also wrapped in MPEG2-TS and HLS containers, MediaConvert inserts SCTE-35 time_signal messages at those points.

For your **Apple HLS** output groups, you can also provide an HLS manifest conditioning XML document. You can then set up your job to condition the manifests for your HLS outputs accordingly.

> **Note**
>
> To put SCTE-35 markers in your MPEG2-TS outputs, in addition to supplying the ESAM XML documents, you must also enable **ESAM SCTE-35** on each output. For more information, see the console procedure following this overview.

About Timecodes in Your ESAM Documents

Specify the insertion points in your XML documents relative to the timing of the final output, after input clipping and stitching. Start your timing from 00:00:00:00, regardless of your timecode settings. Use the following 24-hour format with a frame number: HH:MM:SS:FF.

For example, a job has the following three inputs: a five-minute preroll, a one-hour film, and a five-minute postroll. You use input clipping to clip adjust the final 20 minutes of your one-hour input. So your output with preroll and postroll is 30 minutes long. If you want your first insertion point to appear three minutes into the main content, you would specify it at eight minutes—three minutes after your five-minute preroll.

To include ESAM XML documents in your job settings (console)

1. Open the MediaConvert console at https://console.aws.amazon.com/mediaconvert.
2. Choose **Create new job**.
3. Set up your input, output groups, and outputs for video and audio, as described in Configuring jobs in MediaConvert and Creating outputs.
4. In the **Job** pane on the left, in the **Job settings** section, choose **Settings**.
5. In the **Ad signaling** section, enable **Event signaling and messaging (ESAM)**.
6. For **Signal processing notification XML**, enter your ESAM signaling XML document as text. For an example, see Example ESAM XML signal processing notification.

   > **Note**
   >
   > By default, MediaConvert adds a four-second preroll to the ESAM payload. This might result in MediaConvert placing the SCTE-35 message one segment earlier than the cue marker designates in the HLS manifest. To remove the preroll, set `responseSignalPreroll` to zero. This setting is a child of EsamSettings.

7. If you want to include information about your SCTE-35 markers in your HLS manifests, for **Manifest confirm condition notification XML**, enter your ESAM manifest conditional XML document as text. MediaConvert doesn't include information about your SCTE-35 markers in your DASH manifests.

   To insert SCTE-35 markers in the transport stream in addition to the manifest, set the `dataPassThrough` attribute in your MCC document to `"true"`. If you don't want markers in the transport stream, remove the `dataPassThrough` attributes.

   For an example, see Example ESAM XML Manifest Confirm Condition Notification.
8. For each MPEG2-TS output where you want SCTE-35 markers, enable the markers:

   a. In the **Job** pane on the left, under **Output groups**, **File group**, choose the output.

   b. Confirm that it is an MPEG2-TS output. In the **Output settings** section, make sure that **Container** is set to **MPEG-2 Transport Stream**.

   c. Choose **Container settings**, and then scroll down to find the **PID controls** section.

   d. For **ESAM SCTE-35** choose **Enabled**.
   e. For **SCTE-35 source**, keep the default **None**.

9. Do this step only for any **Apple HLS** output groups in your job.

   If you want to condition your HLS manifest with your ESAM insertion points, follow the procedure in [Including SCTE-35 information in your HLS manifest](). Otherwise, follow these steps to confirm that the following settings are still in their default state:

   a. Make sure that **Manifest confirm condition notification XML**, discussed in a previous step of this procedure, is empty.
   b. For each **Apple HLS** output group in your job, confirm that you have kept **Ad Markers** unchecked.

      i. In the **Job** pane on the left, under **Output groups**, choose **Apple HLS**.
      ii. In the **Apple HLS group settings** section, choose **Advanced**.
      iii. In the **Ad markers** section, clear the **Elemental** and **SCTE-35 enhanced** check boxes.

   c. For each output in your **Apple HLS** output groups, confirm that **SCTE-35 source** is set to **None**:

      i. In the **Job** pane on the left, under **Output groups**, **Apple HLS**, choose an output.
      ii. In the **Output settings** section, choose **Transport stream settings**.
      iii. For **SCTE-35 source**, choose **None**.

To include ESAM XML documents in your jobs settings (API, SDK, AWS CLI)

1. Include the `esam` property and its children at the root of the job settings in your JSON job specification. These properties are shown in the following example.

   a. Include your ESAM signal processing XML specification as a string in the setting `sccXml`.
   b. Optionally, include a manifest confirm condition XML notification document as a string in the setting `mccXml`:

```
"esam": {
  "responseSignalPreroll": 4000,
  "signalProcessingNotification": {
```

```
      "sccXml": "<?xml version=\"1.0\" encoding=\"utf-8\"?>
\n<SignalProcessingNotification ..."
    },
    "manifestConfirmConditionNotification": {
      "mccXml": "<?xml version=\"1.0\" encoding=\"utf-8\" standalone=\"yes\"?>
\n<ns2:ManifestConfirmConditionNotification ... "
    }
```

2. For each M2TS (MPEG2 Transport Stream) output in your job, set your JSON job specification as shown in the following example. Include the property `scte35Esam`. Set `scte35Source` to `NONE`.

```
"outputs": [
      {
        "extension": "m2ts",
        "containerSettings": {
          "container": "M2TS",
          "m2tsSettings": {
            ...

            "scte35Esam": {
              "scte35EsamPid": 508
            },
            ...
            "scte35Source": "NONE"
          }
```

3. If you want to condition your HLS manifests with SCTE-35 information, for each Apple HLS output group in your job, include the following. These settings are shown in the example at the end of this step:

   - Set `scte35Source` to `PASSTHROUGH`.
   - Include `adMarkers` and list one or both of `ELEMENTAL_SCTE35` or `ELEMENTAL` in an array.

     For sample manifests created with each setting selected, see Sample manifest: Elemental ad markers and Sample manifest: SCTE-35 enhanced ad markers.

   If you don't want to condition your HLS manifests with SCTE-35 information, keep the default setting `NONE` for `scte35Source` and don't include `adMarkers`:

```
"outputGroups": [
```

```
    {
      "customName": "apple_hls",
      "outputGroupSettings": {
        "type": "HLS_GROUP_SETTINGS",
        "hlsGroupSettings": {
          "adMarkers": [
            "ELEMENTAL_SCTE35"
          ],
          ...

        }
      },
      "outputs": [
        {
          "extension": "m3u8",
          "nameModifier": "high",
          "outputSettings": {
            "hlsSettings": {
              ...
            }
          },
          "containerSettings": {
            "container": "M3U8",
            "m3u8Settings": {
              ...
              "scte35Source": "PASSTHROUGH"
            }
```

4. Submit your job as usual.

   For information about submitting AWS Elemental MediaConvert jobs programmatically, see Getting started with AWS Elemental MediaConvert using the AWS SDKs or the AWS CLI and Getting started with AWS Elemental MediaConvert using the API.

Example ESAM XML signal processing notification

This ESAM XML block generates two 30-second ad breaks, one at 10 seconds in and the other at 75 seconds in.

```
<?xml version="1.0" encoding="UTF-8"?>
<SignalProcessingNotification xmlns="urn:cablelabs:iptvservices:esam:xsd:signal:1"
 xmlns:sig="urn:cablelabs:md:xsd:signaling:3.0"
```

```
 xmlns:common="urn:cablelabs:iptvservices:esam:xsd:common:1" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" acquisitionPointIdentity="ExampleESAM">

    <common:BatchInfo batchId="1">
      <common:Source xsi:type="content:MovieType" />
    </common:BatchInfo>

    <ResponseSignal acquisitionPointIdentity="ExampleESAM" acquisitionSignalID="1"
 signalPointID="10.00" action="create">
      <sig:NPTPoint nptPoint="10.00"/>
      <sig:SCTE35PointDescriptor spliceCommandType="06">
        <sig:SegmentationDescriptorInfo segmentEventId="1" segmentTypeId="52"
 upidType="9" upid="1" duration="PT30S" segmentNumber="1" segmentsExpected="1"/>
      </sig:SCTE35PointDescriptor>
    </ResponseSignal>
    <ConditioningInfo acquisitionSignalIDRef="1" startOffset="PT10S" duration="PT30S"/>

    <ResponseSignal acquisitionPointIdentity="ExampleESAM" acquisitionSignalID="2"
 signalPointID="40.00" action="create">
      <sig:NPTPoint nptPoint="40.00" />
      <sig:SCTE35PointDescriptor spliceCommandType="06">
        <sig:SegmentationDescriptorInfo segmentEventId="1" segmentTypeId="53"
 upidType="9" upid="2"/>
      </sig:SCTE35PointDescriptor>
    </ResponseSignal>

    <ResponseSignal acquisitionPointIdentity="ExampleESAM" acquisitionSignalID="3"
 signalPointID="75.00" action="create">
      <sig:NPTPoint nptPoint="75.00"/>
      <sig:SCTE35PointDescriptor spliceCommandType="06">
        <sig:SegmentationDescriptorInfo segmentEventId="2" segmentTypeId="52"
 upidType="9" upid="3" duration="PT30S" segmentNumber="2" segmentsExpected="1"/>
      </sig:SCTE35PointDescriptor>
    </ResponseSignal>
    <ConditioningInfo acquisitionSignalIDRef="3" startOffset="PT75S" duration="PT30S"/>

    <ResponseSignal acquisitionPointIdentity="ExampleESAM" acquisitionSignalID="4"
 signalPointID="105.00" action="create">
      <sig:NPTPoint nptPoint="105.00" />
      <sig:SCTE35PointDescriptor spliceCommandType="06">
        <sig:SegmentationDescriptorInfo segmentEventId="2" segmentTypeId="53"
 upidType="9" upid="4"/>
      </sig:SCTE35PointDescriptor>
    </ResponseSignal>
```

```
</SignalProcessingNotification>
```

Example ESAM XML Manifest Confirm Condition Notification

This ESAM manifest confirm condition (MCC) XML document generates HLS manifest cue tags for two 30-second ad breaks. The first begins at 10 seconds in; the second begins at 75 seconds in. These cue manifest tags also contain data that can be used by a downstream client for ad replacement and insertion. Outputs generated with this MCC document contain a SCTE-35 message embedded into the transport stream file at the first ad break, but not at the second one.

Note the `dataPassThrough` attribute on the first ad break. When this attribute is present and set to `"true"`, MediaConvert inserts SCTE-35 markers in the transport stream in addition to the manifest. For any ad break that you want marked only in the manifest, keep out the `dataPassThrough` attribute.

MediaConvert supports manifest conditioning with ESAM only in HLS packages.

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<ns2:ManifestConfirmConditionNotification xmlns:ns2="http://www.cablelabs.com/
namespaces/metadata/xsd/confirmation/2" xmlns="http://www.cablelabs.com/namespaces/
metadata/xsd/core/2" xmlns:ns3="http://www.cablelabs.com/namespaces/metadata/xsd/
signaling/2">
  <ns2:ManifestResponse acquisitionPointIdentity="ExampleESAM" acquisitionSignalID="1"
 duration="PT30S" dataPassThrough="true">
  <ns2:SegmentModify>
      <ns2:FirstSegment>
        <ns2:Tag value="#EXT-X-CUE-OUT:4,SpliceType=VOD_DAI,Action=REPLACE,
 PAID=amazon.com/TEST2014020500000346,Acds=BA" />
      </ns2:FirstSegment>
      <ns2:SpanSegment>
        <ns2:Tag value="#EXT-X-CUE-OUT-CONT:${secondsFromSignal}/4,
 SpliceType=VOD_DAI,Action=REPLACE,PAID=amazon.com/TEST2014020500000346,Acds=BA"
 adapt="true" />
      </ns2:SpanSegment>
      <ns2:LastSegment>
        <ns2:Tag value="#EXT-X-CUE-IN:4,SpliceType=VOD_DAI" locality="after"
 adapt="true" />
      </ns2:LastSegment>
    </ns2:SegmentModify>
 </ns2:ManifestResponse>
```

```
  <ns2:ManifestResponse acquisitionPointIdentity="ExampleESAM"
 acquisitionSignalID="2"></ns2:ManifestResponse>
  <ns2:ManifestResponse acquisitionPointIdentity="ExampleESAM" acquisitionSignalID="3"
 duration="PT30S">
  <ns2:SegmentModify>
      <ns2:FirstSegment>
        <ns2:Tag value="#EXT-X-OUT:4,SpliceType=VOD_DAI,Action=REPLACE,
 PAID=amazon.com/TEST2014020500000347,Acds=BA" />
      </ns2:FirstSegment>
      <ns2:SpanSegment>
        <ns2:Tag value="#EXT-X-OUT-CONT:${secondsFromSignal}/4,
 SpliceType=VOD_DAI,Action=REPLACE,PAID=amazon.com/TEST2014020500000346,Acds=BA"
 adapt="true" />
      </ns2:SpanSegment>
      <ns2:LastSegment>
        <ns2:Tag value="#EXT-X-CUE-IN:4,SpliceType=VOD_DAI" locality="after"
 adapt="true" />
      </ns2:LastSegment>
    </ns2:SegmentModify>
 </ns2:ManifestResponse>
 <ns2:ManifestResponse acquisitionPointIdentity="ExampleESAM"
 acquisitionSignalID="4"></ns2:ManifestResponse>
</ns2:ManifestConfirmConditionNotification>
```

Including SCTE-35 information in your HLS manifest

For outputs in an HLS package, you can have the service include information in the HLS manifest about the SCTE-35 markers that are in each of the outputs.

To include SCTE-35 information in your HLS manifest (console)

1. Open the MediaConvert console at https://console.aws.amazon.com/mediaconvert.
2. Choose **Create new job**.
3. Set up your input, output groups, and outputs for video and audio, following the procedure in Passing through SCTE-35 markers from your input or Specifying SCTE-35 markers using ESAM XML.
4. Enable ad markers in each of your **Apple HLS** output groups.
    a. In the **Job** pane on the left, under **Output groups**, choose **Apple HLS**.
    b. In the **Apple HLS group settings** section, choose **Advanced**.

   c. Choose **Container settings**, then scroll down to find the **PID controls** section.
   d. In the **Ad markers** section, choose one or both of **Elemental** and **SCTE-35 enhanced**.

      For sample manifests created with each setting selected, see Sample manifest: Elemental ad markers and Sample manifest: SCTE-35 enhanced ad markers.

5. If the source of your ad markers is an ESAM document, set these additional settings:

   a. Make sure that you entered XML for **Manifest confirm condition notification XML** during the procedure in Specifying SCTE-35 markers using ESAM XML.

      When you want SCTE-35 markers in the output transport stream as well as in the manifest, use `dataPassThrough` attributes set to `"true"`. When you don't want markers in the transport stream, don't include `dataPassThrough` attributes.
   b. For each output in each of your Apple HLS output groups, set **SCTE-35 source** to **Passthrough**.

      i. In the **Job** pane on the left, under **Output groups**, **Apple HLS**, choose an output.
      ii. In the **Output settings** section, choose **Transport stream settings**.
      iii. For **SCTE-35 source**, choose **Passthrough**.

Sample manifest: Elemental ad markers

To generate the following sample manifest, choose **Apple HLS group settings**, **Advanced**, **Ad markers**, **Elemental**:

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION:12
#EXT-X-MEDIA-SEQUENCE:1
#EXT-X-PLAYLIST-TYPE:VOD
#EXTINF:10.000,
file_60p_1_00001.ts
#EXT-X-CUE-OUT:30.000
#EXTINF:2.000,
file_60p_1_00002.ts
#EXT-X-CUE-OUT-CONT:2.000/30
#EXTINF:12.000,
file_60p_1_00003.ts
```

```
#EXT-X-CUE-OUT-CONT:14.000/30
#EXTINF:12.000,
file_60p_1_00004.ts
#EXT-X-CUE-OUT-CONT:26.000/30
#EXTINF:4.000,
file_60p_1_00005.ts
#EXT-X-CUE-IN
#EXTINF:8.000,
file_60p_1_00006.ts
#EXTINF:12.000,
file_60p_1_00007.ts
#EXTINF:12.000,
file_60p_1_00008.ts
#EXTINF:3.000,
file_60p_1_00009.ts
#EXT-X-CUE-OUT:30.000
#EXTINF:9.000,
file_60p_1_00010.ts
#EXT-X-CUE-OUT-CONT:9.000/30
#EXTINF:12.000,
file_60p_1_00011.ts
#EXT-X-CUE-OUT-CONT:21.000/30
#EXTINF:9.000,
file_60p_1_00012.ts
#EXT-X-CUE-IN
#EXTINF:3.000,
file_60p_1_00013.ts
#EXTINF:12.000,
file_60p_1_00014.ts
#EXTINF:12.000,
file_60p_1_00015.ts
#EXTINF:3.000,
file_60p_1_00016.ts
#EXTINF:9.000,
file_60p_1_00017.ts
#EXTINF:12.000,
file_60p_1_00018.ts
#EXTINF:12.000,
file_60p_1_00019.ts
#EXTINF:12.000,
file_60p_1_00020.ts
#EXTINF:12.000,
file_60p_1_00021.ts
#EXTINF:12.000,
```

```
file_60p_1_00022.ts
#EXTINF:12.000,
file_60p_1_00023.ts
#EXTINF:10.067,
file_60p_1_00024.ts
#EXT-X-ENDLIST
```

Sample manifest: SCTE-35 enhanced ad markers

To generate the following sample manifest, choose **Apple HLS group settings**, **Advanced**, **Ad markers**, **SCTE-35 enhanced**:

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-TARGETDURATION:12
#EXT-X-MEDIA-SEQUENCE:1
#EXT-X-PLAYLIST-TYPE:VOD
#EXTINF:10.000,
file_60p_1_00001.ts
#EXT-OATCLS-SCTE35:/DAnAAAAAAAAAP/wBQb+AA27oAARAg9DVUVJAAAAAX+HCQA0AAE0xUZn
#EXT-X-CUE-OUT:30.000
#EXTINF:2.000,
file_60p_1_00002.ts
#EXT-X-CUE-OUT-CONT:ElapsedTime=2.000,Duration=30,SCTE35=/DAnAAAAAAAAAP/wBQb
+AA27oAARAg9DVUVJAAAAAX+HCQA0AAE0xUZn
#EXTINF:12.000,
file_60p_1_00003.ts
#EXT-X-CUE-OUT-CONT:ElapsedTime=14.000,Duration=30,SCTE35=/DAnAAAAAAAAAP/wBQb
+AA27oAARAg9DVUVJAAAAAX+HCQA0AAE0xUZn
#EXTINF:12.000,
file_60p_1_00004.ts
#EXT-X-CUE-OUT-CONT:ElapsedTime=26.000,Duration=30,SCTE35=/DAnAAAAAAAAAP/wBQb
+AA27oAARAg9DVUVJAAAAAX+HCQA0AAE0xUZn
#EXTINF:4.000,
file_60p_1_00005.ts
#EXT-OATCLS-SCTE35:/DAnAAAAAAAAAP/wBQb+ADbugAARAg9DVUVJAAAAAX+HCQA1AAA3v5+Q
#EXT-X-CUE-IN
#EXTINF:8.000,
file_60p_1_00006.ts
#EXTINF:12.000,
file_60p_1_00007.ts
#EXTINF:12.000,
file_60p_1_00008.ts
```

```
#EXTINF:3.000,
file_60p_1_00009.ts
#EXT-OATCLS-SCTE35:/DAnAAAAAAAAAP/wBQb+AGb/MAARAg9DVUVJAAAAAn+HCQA0AALMua1L
#EXT-X-CUE-OUT:30.000
#EXTINF:9.000,
file_60p_1_00010.ts
#EXT-X-CUE-OUT-CONT:ElapsedTime=9.000,Duration=30,SCTE35=/DAnAAAAAAAAAP/wBQb+AGb/
MAARAg9DVUVJAAAAAn+HCQA0AALMua1L
#EXTINF:12.000,
file_60p_1_00011.ts
#EXT-X-CUE-OUT-CONT:ElapsedTime=21.000,Duration=30,SCTE35=/DAnAAAAAAAAAP/wBQb+AGb/
MAARAg9DVUVJAAAAAn+HCQA0AALMua1L
#EXTINF:9.000,
file_60p_1_00012.ts
#EXT-OATCLS-SCTE35:/DAnAAAAAAAAAP/wBQb+AJAyEAARAg9DVUVJAAAAAn+HCQA1AABStd4A
#EXT-X-CUE-IN
#EXTINF:3.000,
file_60p_1_00013.ts
#EXTINF:12.000,
file_60p_1_00014.ts
#EXTINF:12.000,
file_60p_1_00015.ts
#EXTINF:3.000,
file_60p_1_00016.ts
#EXTINF:9.000,
file_60p_1_00017.ts
#EXTINF:12.000,
file_60p_1_00018.ts
#EXTINF:12.000,
file_60p_1_00019.ts
#EXTINF:12.000,
file_60p_1_00020.ts
#EXTINF:12.000,
file_60p_1_00021.ts
#EXTINF:12.000,
file_60p_1_00022.ts
#EXTINF:12.000,
file_60p_1_00023.ts
#EXTINF:10.067,
file_60p_1_00024.ts
#EXT-X-ENDLIST
```

Enabling ad avail blanking

You can enable ad avail blanking to remove video content, remove any captions, and mute audio during the portions of the output that are marked as available for ads (ad avails).

You set up SCTE-35 markers in each output individually, but you enable or disable ad avail blanking for every output in the job. To use ad avail blanking, you have to both set up SCTE-35 markers and enable ad avail blanking, as described in the following procedure.

To enable ad avail blanking (console)

1. Open the MediaConvert console at https://console.aws.amazon.com/mediaconvert.
2. Choose **Create new job**.
3. Set up your input, output groups, and outputs for video and audio, following the procedure in Passing through SCTE-35 markers from your input or Specifying SCTE-35 markers using ESAM XML.
4. In the left navigation pane, under **Job settings**, choose **Settings**.
5. Under **Global processors**, enable **Ad avail blanking**.
6. Optionally, under **Blanking image**, provide a URI to an image input file that is stored in Amazon S3 or on an HTTP(S) server. For Amazon S3 inputs, you can specify the URI directly or choose **Browse** to select from your Amazon S3 buckets. For HTTP(S) inputs, provide the URL to your input video file. For more information, see HTTP input requirements.

   If you specify an image here, the service inserts the image on all video frames inside the ad avail. If you don't specify an image, the service uses a black slate instead.

   Blanking images must be .png or .bmp files that are the same size or smaller, in pixels, as the output video resolution.

Setting up timecodes

AWS Elemental MediaConvert manages transcoded video frames by their timecode. The service uses the timecodes from the input and output timelines that it constructs to line up the elements of your output assets. For information about which features are affected by each type of timeline, and about how timelines work, see How MediaConvert uses timelines to assemble jobs.

There are three distinct groups of timecode settings, located in three different places on the console:

- Input timecode settings

  The input setting **Timecode source** affects the input timeline.

- Job-wide timecode configuration

  The **Timecode configuration** settings under **Job settings** affect the output timeline.

- Output timecode settings

  The timecode settings under **Output** determine whether and how timecode information appears in each output. These settings affect only what is included in the outputs; they don't determine what timecodes are.

To provide frame accuracy for video inputs, MediaConvert uses timecodes that specify frames by frame number, not by millisecond. All timecodes are in the following 24-hour format with a frame number: HH:MM:SS:FF. For drop frame, MediaConvert uses a semicolon before the frame number: HH:MM:SS;FF.

When you specify an input clip for an audio-only input, the last numbers in the timecode that you enter correspond to hundredths of a second. For example, 00:00:30:75 is the same as 30.75 seconds.

Topics

Adjusting the input timeline with the input timecode source

The value for **Timecode source** that you specify in an input's settings affects the input timeline for that input. For information about which features are affected by the input timeline, see Input timelines.

To adjust the input Timecode source setting (console)

1. On the **Create job** page, in the **Job** pane on the left, choose an input.

2. Under **Video selector**, **Timecode source**, specify whether MediaConvert reads timecodes from the input or generates them. MediaConvert can generate timecodes starting from zero or from a starting timecode that you specify. Here are the options for **Timecode source**:

   - **Embedded**: The service uses any timecodes embedded in the input video. This is the default value.

     > **Note**
     >
     > Don't choose this value unless your input video has embedded timecodes.

   - **Start at 0**: The service sets the timecode of the first frame of the input to 00:00:00:00.
   - **Specified start**: The service sets the timecode of the first frame of the input to the value that you specify in the setting **Start timecode**.

   Regardless of the source, timecodes are in the following 24-hour format with a frame number: HH:MM:SS:FF.

To adjust the input `TimecodeSource` (Timecode source) setting (API, SDK, and AWS CLI)

- In your JSON job specification, set a value for TimecodeSource, located in `Settings`, `Inputs`.

  Choose a value for `TimecodeSource` as follows:

  - **EMBEDDED**: The service uses any timecodes embedded in the input video. This is the default value.

    > **Note**
    >
    > Don't choose this value unless your input video has embedded timecodes.

  - **ZEROBASED**: The service sets the timecode of the first frame of the input to 00:00:00:00.
  - **SPECIFIEDSTART**: The service sets the timecode of the first frame of the input to the value that you specify in the setting **Start timecode**.

Adjusting the output timeline with the job-wide timecode configuration

The values that you specify for the job-wide **Timecode configuration** settings affect the output timeline. For information about which features are affected by the output timeline, see Output timeline .

To adjust the job-wide timecode configuration (console)

1. On the **Create job** page, in the **Job** pane on the left, choose **Settings**.
2. In the **Timecode configuration** section, for **Source**, choose one of the following values:

   - **Embedded**: The service uses any timecodes that are embedded in the video.
   - **Start at 0**: The service ignores any embedded timecodes and assigns the first video frame the timecode 00:00:00:00 (HH:MM:SS:FF).
   - **Specified start**: The service ignores any embedded timecodes and assigns the first video frame the value that you provide for **Start Timecode**.

     The **Start Timecode** field appears when you choose **Specified start**.

   If you use the API or an SDK, you can find this setting in the JSON file of your job. The setting name is `Source`, located inside `Settings`, `TimecodeConfig`.

   If you don't choose a value for **Source**, the service defaults to **Embedded**.

   > **Note**
   >
   > If your input video doesn't have embedded timecodes and you set **Source** to **Embedded** or keep **Source** unspecified, your output won't have timecodes. This means that features that require a timecode-based start time, such as sidecar captions and graphic overlays, won't appear in your output.

3. Set a value for **Anchor Timecode**.

   If you use an editing platform that relies on an anchor timecode, use **Anchor timecode** to specify a point at which the input and output frames have the same timecode. Use the following 24-hour format with a frame number: HH:MM:SS:FF. This setting ignores frame rate conversion.

The system behavior for **Anchor timecode** varies depending on your setting for **Source**:

- If you choose **Start at 0** for **Source**, the anchor frame is the timecode that you provide in **Anchor timecode**, counting from 00:00:00:00.

  For example, if you set **Anchor timecode** to 01:00:05:00, the anchor frame is one hour and five seconds into the video.
- If you choose **Embedded** for **Source**, the anchor frame is the timecode that you provide in **Anchor timecode**, counting from the first embedded timecode.

  For example, if your embedded timecodes start at 01:00:00:00 and you set **Anchor timecode** to 01:00:05:00, the anchor frame is five seconds into the video.
- If you choose **Specified start** for **Source**, the anchor frame is the timecode that you provide in **Anchor timecode**, counting from the timecode that you specify for the first frame.

  For example, if you specify 00:30:00:00 as your start timecode and you set **Anchor timecode** to 01:00:05:00, the anchor frame is thirty minutes and five seconds into the video.

If you use the API or an SDK, you can find this setting in the JSON file of your job. The setting name is `Anchor`, located in `Settings`, `TimecodeConfig`.

If you don't set a value for **Anchor timecode**, the service doesn't use any anchor timecode.

4. Under **Timestamp offset**, provide a date. This setting applies only to outputs that support a program-date-time stamp. Use **Timestamp offset** to overwrite the timecode date without affecting the time and frame number. This setting has no effect unless you also include the program-date-time stamp in the output.

   If you use the API or an SDK, you can find this setting in the JSON file of your job. The setting name is `TimestampOffset`, located in `Settings`, `TimecodeConfig`.

To adjust the job-wide timecode configuration (API, SDK, and AWS CLI)

1. In your JSON job specification, set a value for Source, located inside `Settings`, `TimecodeConfig`. Choose one of the following values:

   - **EMBEDDED**: The service uses any timecodes that are embedded in the video.

- **ZEROBASED**: The service ignores any embedded timecodes and assigns the first video frame the timecode 00:00:00:00 (HH:MM:SS:FF).
- **SPECIFIEDSTART**: The service ignores any embedded timecodes and assigns the first video frame the value that you provide for **Start Timecode**.

  The **Start Timecode** field appears when you choose **Specified start**.

If you don't choose a value for **Source**, the service defaults to **Embedded**.

> **Note**
>
> If your input video doesn't have embedded timecodes and you set **Source** to **Embedded** or keep **Source** unspecified, your output won't have timecodes. This means that features that require a timecode-based start time, such as sidecar captions and graphic overlays, won't appear in your output.

2. Optional. In your JSON job specification, set a value for `Anchor`, located in `Settings`, `TimecodeConfig`.

   If you use an editing platform that relies on an anchor timecode, use `Anchor` to specify a point at which the input and output frames have the same timecode. Use the following 24-hour format with a frame number: HH:MM:SS:FF. This setting ignores frame rate conversion.

   The system behavior for `Anchor` varies depending on your setting for `Source`:

   - If you choose `ZEROBASED` for `Source`, the anchor frame is the timecode that you provide in `Anchor`, counting from 00:00:00:00.

     For example, if you set `Anchor` to 01:00:05:00, the anchor frame is one hour and five seconds into the video.
   - If you choose `EMBEDDED` for `Source`, the anchor frame is the timecode that you provide in `Anchor`, counting from the first embedded timecode.

     For example, if your embedded timecodes start at 01:00:00:00 and you set `Anchor` to 01:00:05:00, the anchor frame is five seconds into the video.
   - If you choose `SPECIFIEDSTART` for `Source`, the anchor frame is the timecode that you provide in `Anchor`, counting from the timecode that you specify for the first frame.

For example, if you specify 00:30:00:00 as your start timecode and you set `Anchor` to 01:00:05:00, the anchor frame is thirty minutes and five seconds into the video.

3. Optional. In your JSON job specification, set a value for `TimestampOffset`, located in `Settings`, `TimecodeConfig`. Specify the date in the following format: `YYYY-MM-DD`. For example, `2008-06-26`.

   This setting applies only to outputs that support a program-date-time stamp. Use **Timestamp offset** to overwrite the timecode date without affecting the time and frame number. This setting has no effect unless you also include the program-date-time stamp in the output.

Putting timecodes in your outputs

There are two timecode-related settings that you can adjust differently for each output: **Timecode insertion** and **Timecode burn-in**.

Inserting timecode metadata

The **Timecode insertion** setting determines whether a given output has timecodes embedded in its metadata. MediaConvert automatically puts this information in the appropriate place, depending on the output codec. For MPEG-2 and QuickTime codecs, such as Apple ProRes, the service inserts the timecodes in the video I-frame metadata. For H.265 (HEVC) and H.264 (AVC), the service inserts the timecodes in the supplemental enhancement information (SEI) picture timing message.

To include timecode metadata in an output (console)

1. On the **Create job** page, in the **Job** pane on the left, choose an output.
2. Under **Stream settings**, **Timecode insertion**, choose **Insert** to include timecode metadata. Choose **Disabled** to omit timecode metadata.

To include timecode metadata in an output (API, SDK, and AWS CLI)

- In your JSON job specification, set a value for [TimecodeInsertion](), located in `Settings`, `OutputGroups`, `Outputs`, `VideoDescription`.

  Use `PIC_TIMING_SEI` to include timecode metadata. Use `DISABLED` to omit timecode metadata.

Burning in timecodes on the video frames

The **Timecode burn-in** setting determines whether a given output has visible timecodes inscribed into the video frames themselves. The timecodes are not an overlay, but rather a permanent part of the video frames.

To burn in timecodes in an output (console)

1. On the **Create job** page, in the **Job** pane on the left, choose an output.
2. Under **Stream settings**, **Preprocessors**, choose **Timecode burn-in**.
3. Optionally, provide values for the **Prefix**, **Font size**, and **Position** settings. Even if you don't provide these values, timecodes are burned into your output using these default values:

   - **Prefix**: no prefix
   - **Font size**: **Extra Small (10)**
   - **Position**: **Top Center**

   For details about each of these settings, choose the **Info** link next to **Timecode burn-in**.

To burn in timecodes in an output (API, SDK, and AWS CLI)

1. In your JSON job specification, include the setting TimecodeBurnin. `TimecodeBurnin` is located in `Settings`, `OutputGroups`, `Outputs`, `VideoDescription`, `VideoPreprocessors`.
2. Optionally, provide values for the settings that are children of `TimecodeBurnin`. If you don't provide these values, timecodes are burned into your output using these default values:

   - `Prefix`: *no prefix*
   - `FontSize`: `10`
   - `Position`: `TOP_CENTER`

Generating black video

This guide shows you how to generate black video with AWS Elemental MediaConvert. To generate black video, you can add an input and include **Video generator**, or create a video output from an input that doesn't have video.

Workflows to consider when generating black video:

- Insert black video at the beginning of your content.
- Insert black video between two inputs.
- Insert black video at the end of your content.
- Create a black video track for an audio-only or captions-only input.
- Any previous combination.

How to generate black video

In the following steps, you will include **Video generator** to generate a black video. When you do, also specify a numerical value for **Duration** in milliseconds from `50` to `86400000`. This creates a video input with black frames for that duration without an audio track.

Alternatively, MediaConvert automatically creates black video when the following conditions are met:

- Your input doesn't have video. Examples include:
  - Audio-only inputs
  - Captions-only inputs (in sidecar formats)
- Your output includes a video track.

In the previous examples, the duration of the black video that you generate will match the duration of the input audio or captions.

Generate black video by adding an input with Video generator specified.

1. In the **Input** pane, toggle on **Video generator**.
2. Specify a value for **Duration** in milliseconds.
3. After defining the rest of your job settings, choose **Create**.

Create a black video track for an audio-only input.

1. In the **Input** pane, keep **Input file URL** blank.
2. Under **Audio selectors**, **Audio Selector 1**, switch on **External file**.

3. Enter the URL of your audio input.
   - If your input has both audio and video, MediaConvert ignores the input video.
4. Under **Audio selectors**, specify any other required input audio settings.
5. See Step 3: Create output groups and Step 4: Create outputs to set up your outputs.
   a. You must include a video track in your output.
   b. You must include an audio track in your output, with **Audio source** set to the **Audio selector** specified from step 2, shown previously.
6. After defining the rest of your job settings, choose **Create**.
7. MediaConvert automatically creates black video with the same duration as the input audio selector.

Create a black video track for a captions-only input.

1. In the **Input** pane, keep **Input file URL** blank.
2. Next to **Captions selectors**, choose **Add captions selector**.
3. In **Captions Selector 1**, under **Source**, choose a sidecar captions format.
   - Non-sidecar captions formats are not supported.
4. Enter the URL of your captions input.
5. See Step 3: Create output groups and Step 4: Create outputs to set up your outputs.
   a. You must include a video track in your output.
   b. You must include a captions track in your output, with **Captions source** set to the **Captions selector** specified from step 2 above.
6. After defining the rest of your job settings, choose **Create**.
7. MediaConvert automatically creates black video, with the same duration as the input **Captions selector**.

Video Generator FAQ

Q: What if my job generates black video and I set my output frame rate to Follow source?

If your job does not include any other inputs, there is no input frame rate for MediaConvert to follow. You must specify an output frame rate.

If your job includes any other inputs with video, MediaConvert uses the frame rate from the first video input.

Q: What if my job generates black video I don't define an output resolution?

If your job does not include any other inputs, there is no input resolution for MediaConvert to follow. Specify an output resolution.

If your job includes any other inputs with video, MediaConvert uses the resolution from the first video input.

Feature restrictions for Video generator

The following feature is unavailable when you generate black video:

- Accelerated transcoding

Using video overlays

With the **Video overlays** feature, you can replace video frames from your input with frames from another video. Video overlays do not affect audio or captions from your underlying input, which continue uninterrupted. This is useful when you want to quickly replace video content without spending time configuring multiple inputs or input clips. Common workflows might be to blank out content captured during a live broadcast, to blank out content that you don't have the rights to, or to provide alternate footage for a scene.

When you add a video overlay, you specify its duration by providing the timecode that it starts and ends within the underlying input video. If your video overlay would end before the end timecode, your video overlay will repeat to meet the total duration.

Additionally, you can specify one or more clips to use from your video overlay file using **Input clips**. You can include up to 99 video overlays and 99 input clips in each video overlay.

Topics

- How to add a video overlay
- Video overlay feature restrictions

How to add a video overlay

Video overlays are specific to individual video inputs. For example, if your job settings include three video inputs, you can add an individual video overlay to **Input 1**, **Input 2**, and/or **Input 3**. You cannot however add a single video overlay that spans across all three inputs.

MediaConvert console

To add a video overlay by using the MediaConvert console:

1. Open the Create job page in the MediaConvert console.
2. Specify an input video. For more information, see *Step 1: Input files* in Configuring jobs.
3. In the settings for your input, next to **Video overlays**, choose **Add overlay**.
4. Specify an Input file URI.
5. Specify a **Start timecode**.
6. Specify an **End timecode**.
7. Optionally add any **Input clips** for your video overlay.

API, SDK, or AWS Command Line Interface (AWS CLI)

To specify a video overlay using the API, SDK, or AWS Command Line Interface (AWS CLI), include `VideoOverlays` in your job settings JSON.

The following is an excerpt of a job settings JSON that specifies a video overlay at timecode `00:10:00:00` for one minute from a clipped input:

```
{
  "Settings": {
    "Inputs": [
      {
        "VideoOverlays": [
          {
            "Input": {
              "FileInput": "s3://DOC-EXAMPLE-BUCKET/overlay.mp4",
              "InputClippings": [
                {
                  "StartTimecode": "00:10:00:00",
                  "EndTimecode": "00:11:00:00"
```

```
                }
              ],
              "TimecodeSource": "EMBEDDED"
            },
            "StartTimecode": "00:01:00:00",
            "EndTimecode": "00:02:00:00"
          }
        ]
      }
    ]
  }
}
```

Video overlay feature restrictions

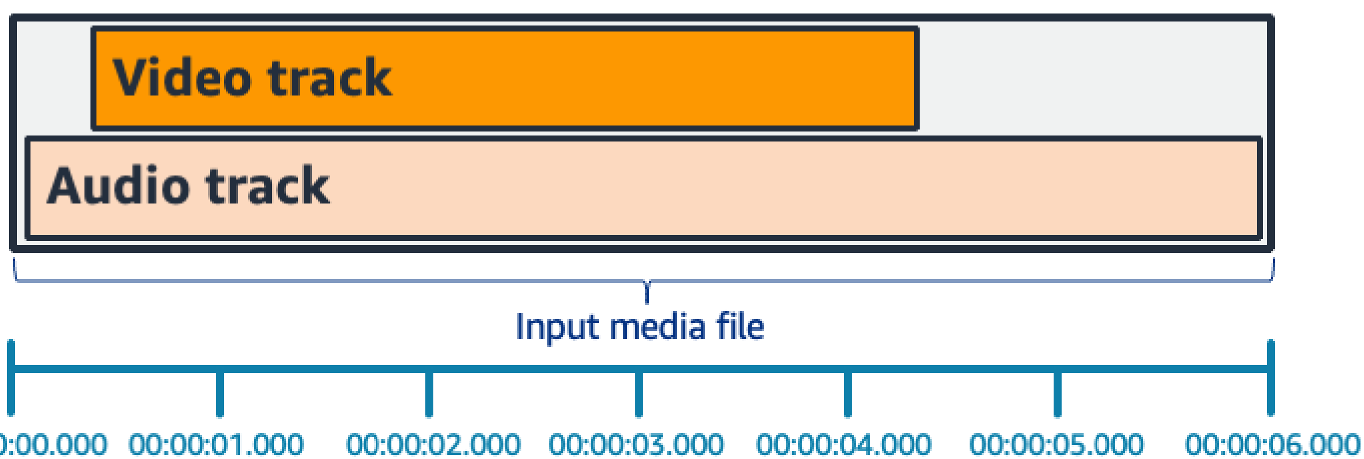
Video overlay files inherit the color information from their underlying input video. For example, if your input video uses a Rec. 601 color space, MediaConvert assumes your video overlay also uses a Rec. 601 color space.

For consistency in color and formatting in your output video image, we recommend that you specify video overlay files with similar characteristics as their underlying input videos.

Padding video with black video frames

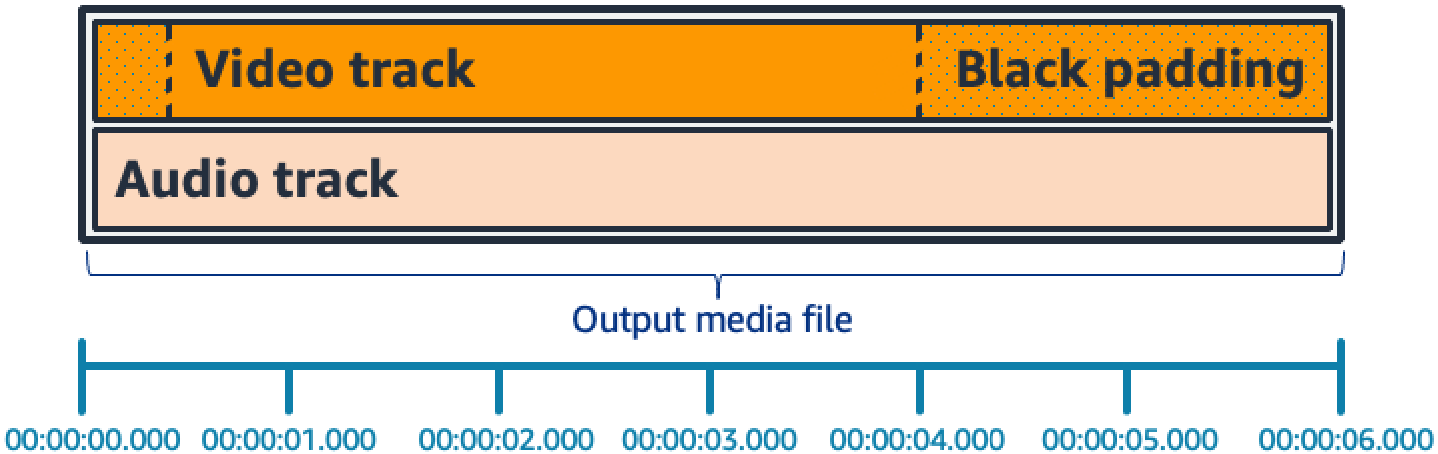
AWS Elemental MediaConvert can pad video tracks with black video frames so that video and audio durations align.

Input media files might have audio and video tracks with different durations. In the following horizontal bar graph, the audio track starts before the video track and ends after it.



You can use **Pad video** to lengthen the video track. In the **Input** pane, under **Video selector**, choose **Video correction** and set **Pad video** to `Black`. Depending on your input, this generates black video frames at the beginning, end, or both the beginning and end, of your video.

In the following horizontal bar graph, black frames fill in the beginning and end of the video to match the length of the audio.



Pad video FAQ

Q: Since video frames and audio samples have different rates, will AWS Elemental MediaConvert ever generate partial video frames?

MediaConvert generates full video frames.

If you have a downstream workflow that is sensitive to very small duration differences between video and audio, set **Audio duration** to `Match video duration` in your output video settings.

Q: What if the starting presentation timestamps (PTS) in my input video and audio tracks are different?

If your input video starts at 10 seconds and has a 30 second duration, and your input audio starts at 0 seconds and also has a 30 second duration, then 10 seconds of black video will be added to the beginning.

Q: What if the audio in my input starts *after* the video? Or the audio ends before the video?

Audio silence is inserted at the beginning or end to align with the start or end of the video.

Q: Can I use input clipping instead to align audio and video durations?

Yes, input clipping can help accomplish the same task of aligning audio and video durations. But keep in mind that input clipping also removes audio or video content from the input.

Feature restrictions for Pad video

The following feature is unavailable when you pad with black video:

- Accelerated transcoding

Using video rotation in AWS Elemental MediaConvert

For most inputs, you can choose how AWS Elemental MediaConvert rotates your video. You can either specify the rotation or set the rotation to automatic. Automatic rotation uses any rotation metadata contained in the input files. Some cameras, often those in smartphones, record this rotation metadata when you turn the camera before you begin recording your video. This rotation metadata, sometimes referred to as rotation atoms or boxes, provides rotation metadata to the player device that is used for viewing the video. MediaConvert can automatically detect this rotation metadata and rotate your video during transcoding so that it appears correctly on all players.

> **Tip**
>
> If your video outputs are rotated in a way that you don't expect, a likely cause is that your input video has rotation metadata but your job settings don't specify that the service should use it. Try setting **Rotate** to **Automatic**. Find this setting on the **Create job** page, under **Input**, in the **Video selector** section.

Input file requirements

You can use rotation for inputs that have the following video characteristics:

- Progressive video
- Chroma subsampling scheme 4:2:2 or 4:2:0

Topics

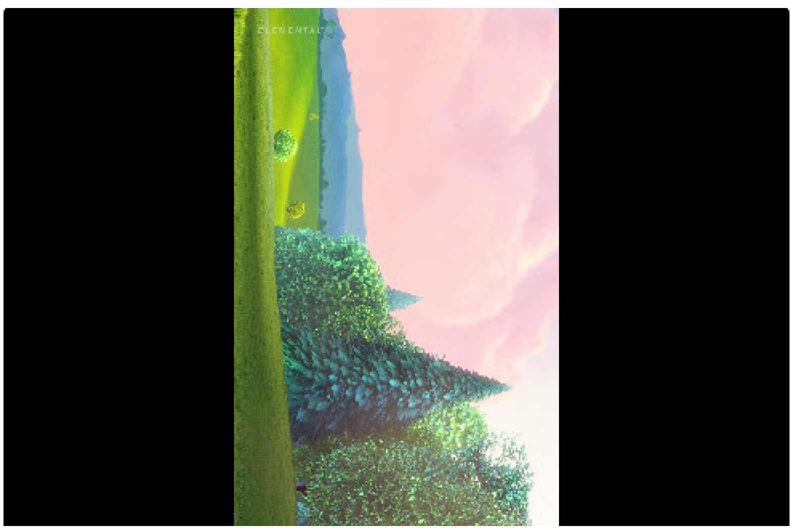
- [Specified rotation]

- Automatic rotation

Specified rotation

When you specify the rotation for your input, AWS Elemental MediaConvert rotates the video from your input clockwise the amount that you specify. This rotation applies to all outputs in the job. You can rotate clockwise by 90, 180, or 270 degrees. The following image shows a video output from a job that specifies a 90-degree rotation.

> **Note**
>
> AWS Elemental MediaConvert doesn't pass through rotation metadata. Regardless of how you set **Rotate**, job outputs don't have rotation metadata.



To specify the rotation of your video

1. On the **Create job** page, in the **Job** pane on the left, in the **Inputs** section, choose the input that you want to rotate.
2. In the **Video selector** section on the left, for **Rotate**, choose the amount of clockwise rotation that you want.

If you use the API or an SDK, you can find this setting in the JSON file of your job. The setting name is `rotate`. Find the rotate property in the AWS Elemental MediaConvert API Reference.

> **Note**
>
> AWS Elemental MediaConvert doesn't rotate images and motion images that you overlay. If you use the image inserter feature or the motion image inserter feature with the rotate feature, rotate your overlay before you upload it. Specify the position of your overlays as you want them to appear on the video after rotation.

Automatic rotation

If your video has embedded rotation metadata, AWS Elemental MediaConvert can detect it and automatically rotate your video content so that it's oriented correctly in your outputs.

> **Note**
>
> AWS Elemental MediaConvert doesn't pass through rotation metadata. Regardless of how you set **Rotate**, job outputs don't have rotation metadata.

Additional input file requirements for automatic rotation

In addition to the general input restrictions for the rotate feature, to use automatic rotation your input file must conform to these limitations:

- Input container: .mov or .mp4
- Rotation metadata specifying 90, 180, or 270-degree rotation

  If your rotation metadata is within one degree less or more than the values listed here, the service will round to a supported value.

> **Note**
>
> If your input file has rotation metadata that specifies a rotation other than those listed here, the service defaults to no rotation.

To enable automatic rotation

1. Check that your input container is .mov or .mp4 and that your input has rotation metadata.
2. On the **Create job** page, in the **Job** pane on the left, in the **Inputs** section, choose the input that has rotation metadata.
3. In the **Video selector** section on the left, for **Rotate**, choose **Automatic**.

> **Note**
>
> AWS Elemental MediaConvert doesn't rotate images and motion images that you overlay. If you use the image inserter feature or the motion image inserter feature with the rotate feature, rotate your overlay before you upload it. Specify the position of your overlays as you want them to appear on the video after rotation.

Video scaling behavior and aspect ratios

You can choose how AWS Elemental MediaConvert scales your input video image to the output resolution that you specify by using **Scaling behavior**.

The following tables show how **Scaling behavior**, **Scaling padding**, **Height**, and **Width** interact to change the output image.

Topics

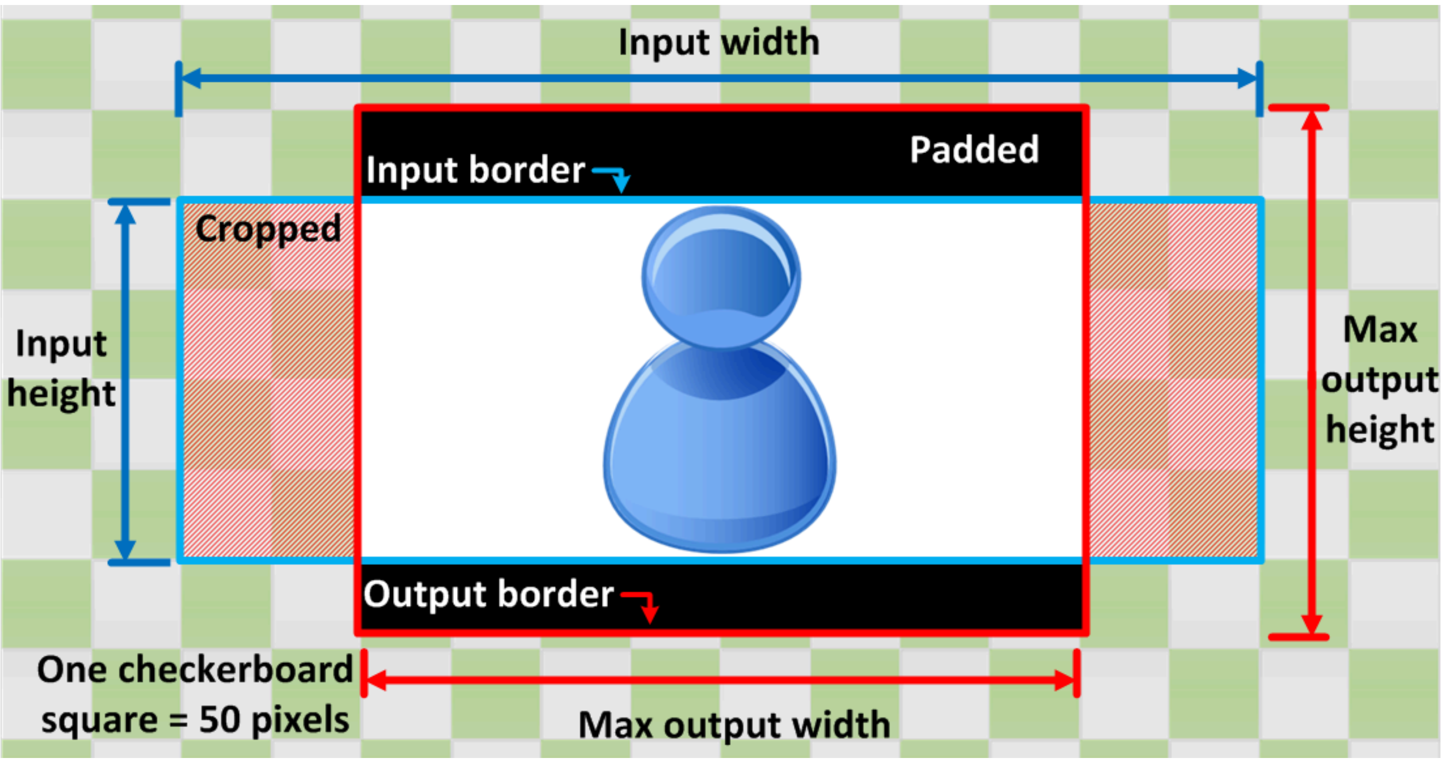
Default (Fit with padding)

If you choose **Default (Fit with padding)** for your **Scaling behavior**, MediaConvert scales your video image to your output resolution. Then, if your input resolution has a different aspect ratio than your output resolution, MediaConvert pads your video image until it matches the dimensions of your output resolution.

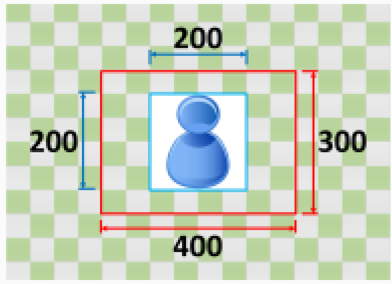
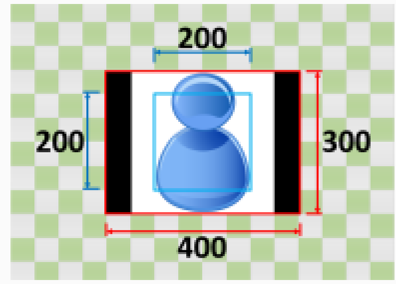
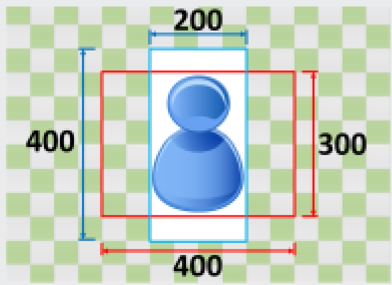
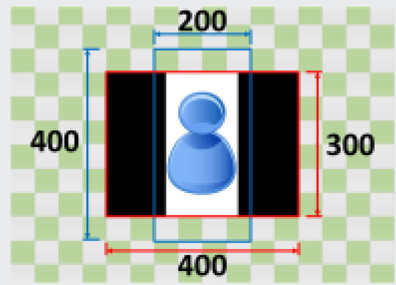
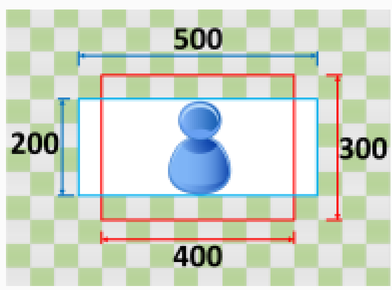
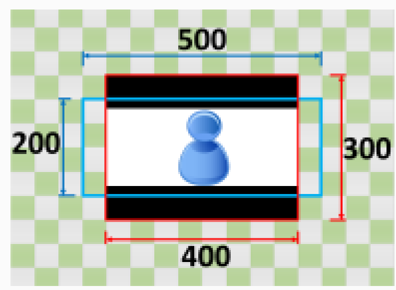
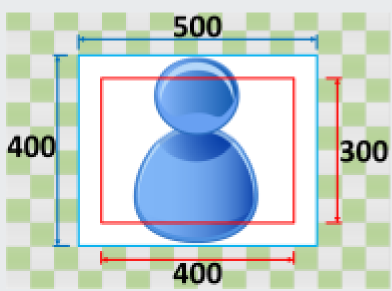
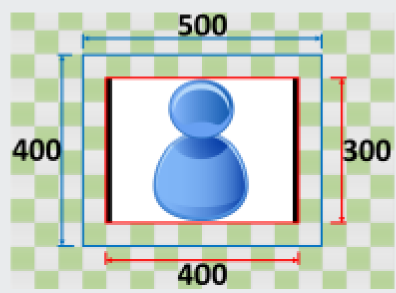
For example, if your input file is `1280` pixels by `720` pixels and you specify an output resolution that is `640` pixels by `480` pixels, MediaConvert reduces the image size to `640x360` and then pads the top and bottom of the image so that the final video resolution is `640x480`. MediaConvert does not use cropping for the **Default** scaling behavior.

Key

The following key graphic shows input and output image width, height, cropping, and padding. In this key graphic, input dimensions (on the top and left) are in blue and output dimensions (on the right and bottom) are in red.



The following table shows example image scaling behavior when your input and output resolutions differ. For details about to read the images, reference the previous key graphic.

Condition	Input	Output
Input width less than output width Input height less than output height		
Input width less than output width Input height greater than output height		
Input width greater than output width Input height less than output height		
Input width greater than output width Input height greater than output height		

Stretch to output

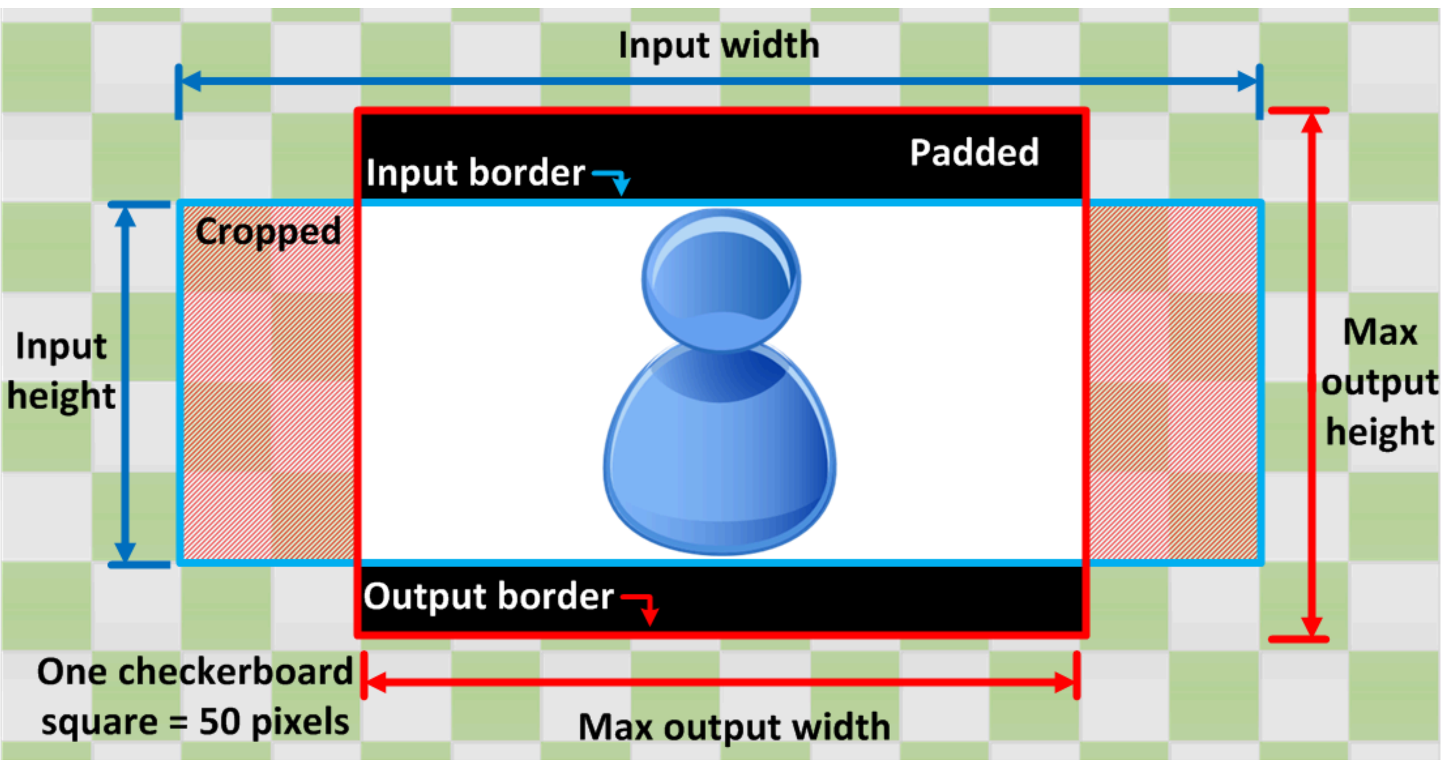
If you choose **Stretch to output** for your **Scaling behavior**, MediaConvert stretches or shrinks your video image to your output resolution.

For example, if your input image is `200` pixels by `200` pixels and you want an output resolution that is `300` pixels by `400` pixels, MediaConvert increases the size of your input image to `300` pixels by `400` pixels, distorting the proportions of your output image. MediaConvert does not use padding or cropping for the **Stretch to output** scaling behavior.

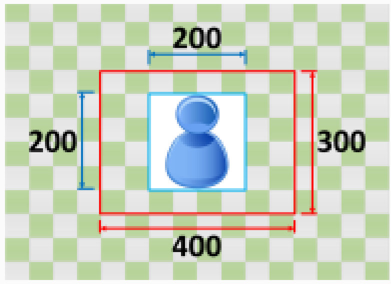
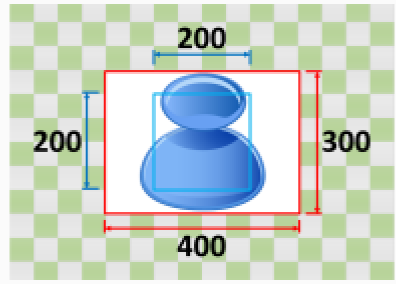
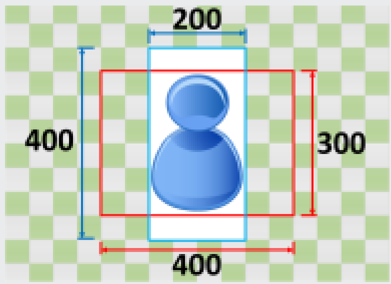
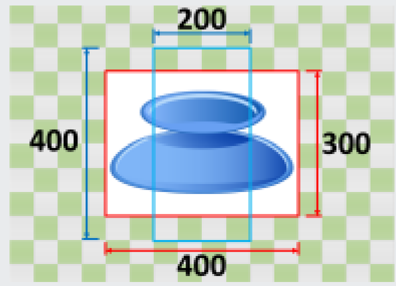
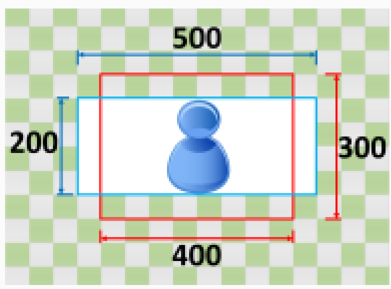
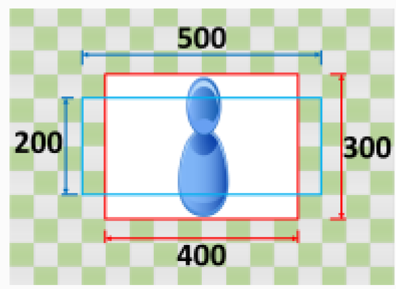
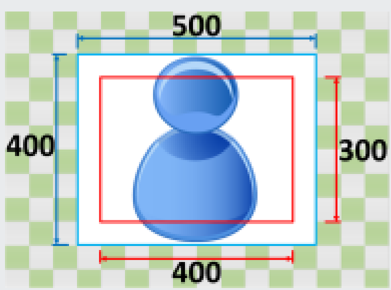
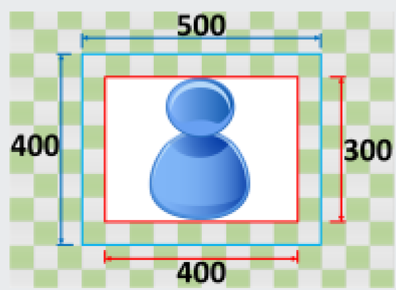
> **Important**
>
> If your input resolution has a different aspect ratio than your output resolution, your output image will be distorted when compared to your input image.

Key

The following key graphic shows input and output image width, height, cropping, and padding. In this key graphic, input dimensions (on the top and left) are in blue and output dimensions (on the right and bottom) are in red.



The following table shows example image scaling behavior when your input and output resolutions differ. For details about to read the images, reference the previous key graphic.

Condition	Input	Output
Input width less than output width Input height less than output height		
Input width less than output width Input height greater than output height		
Input width greater than output width Input height less than output height		
Input width greater than output width Input height greater than output height		

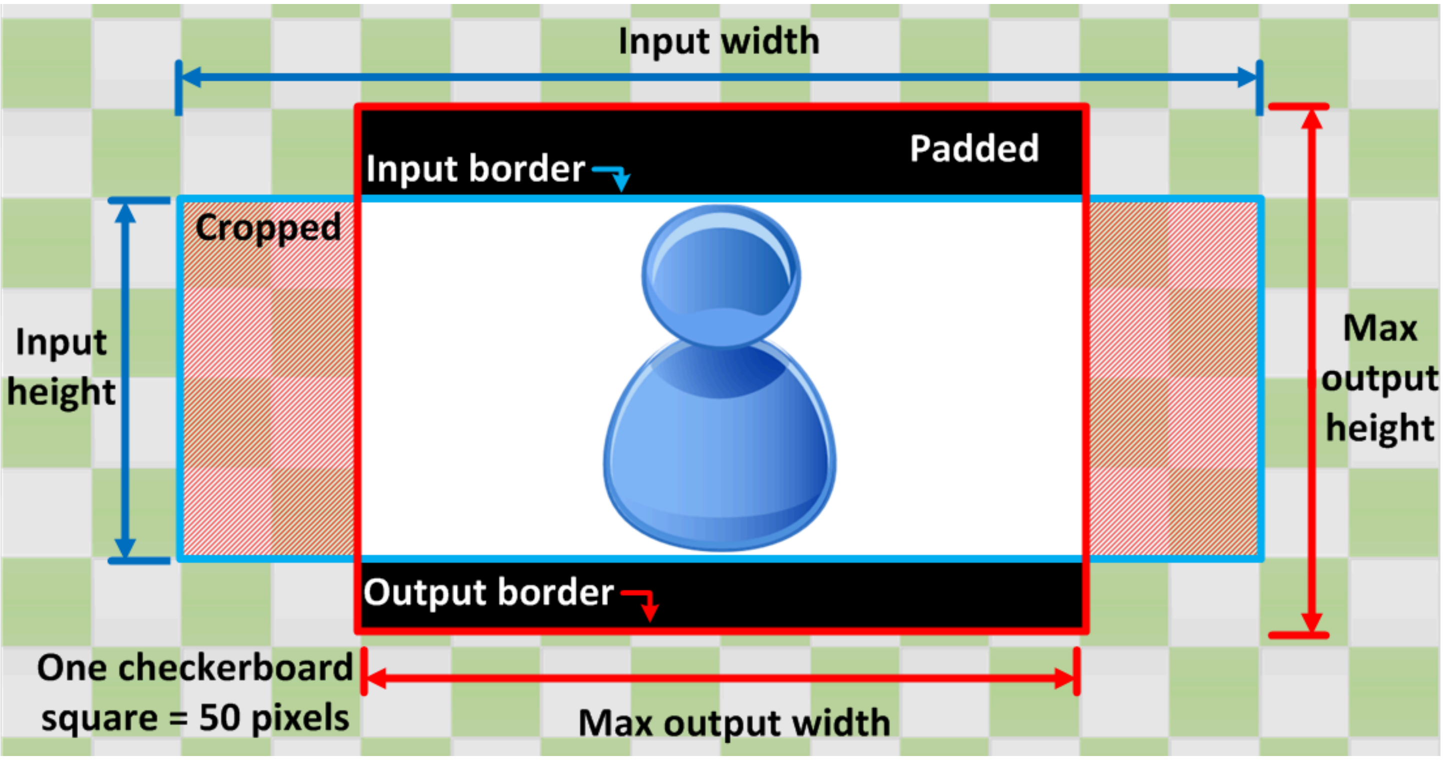
Fit

If you choose **Fit** for your **Scaling behavior**, MediaConvert scales your input image until it fits inside the dimensions of your output resolution, without exceeding the dimensions of your output resolution.

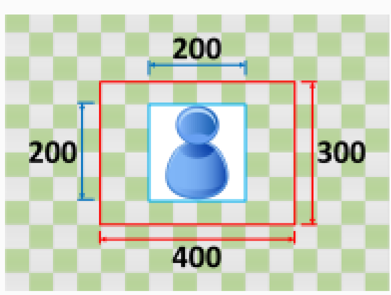
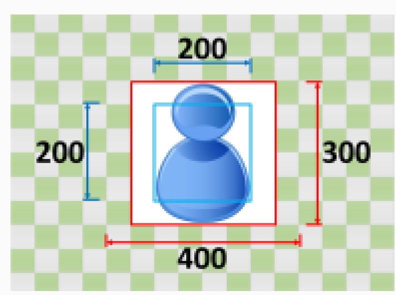
For example, if your input file is `200` pixels by `200` pixels and you want an output resolution that is `300` pixels by `400` pixels, MediaConvert increases the image to `300` pixels by `300` pixels. MediaConvert does not add padding to your output when you choose **Fit**.

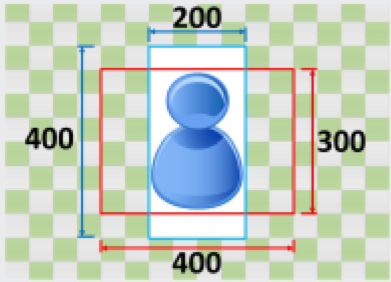
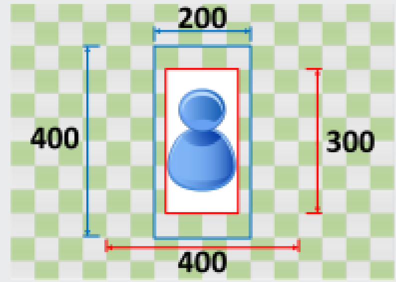
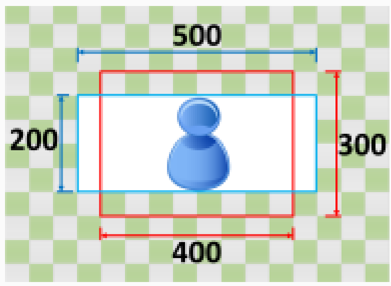
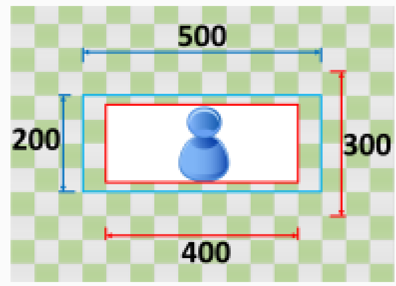
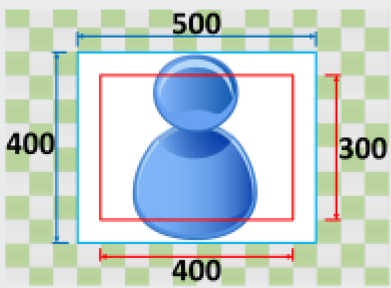
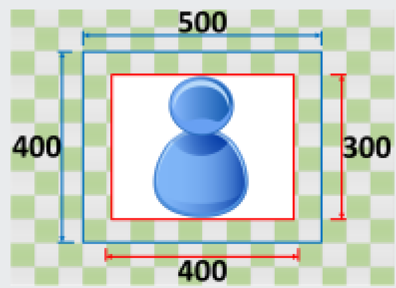
Key

The following key graphic shows input and output image width, height, cropping, and padding. In this key graphic, input dimensions (on the top and left) are in blue and output dimensions (on the right and bottom) are in red.



The following table shows example image scaling behavior when your input and output resolutions differ. For details about to read the images, reference the previous key graphic.

Condition	Input	Output
Input width less than output width Input height less than output height		

Condition	Input	Output
Input width less than output width Input height greater than output height		
Input width greater than output width Input height less than output height		
Input width greater than output width Input height greater than output height		

Fit without upscaling

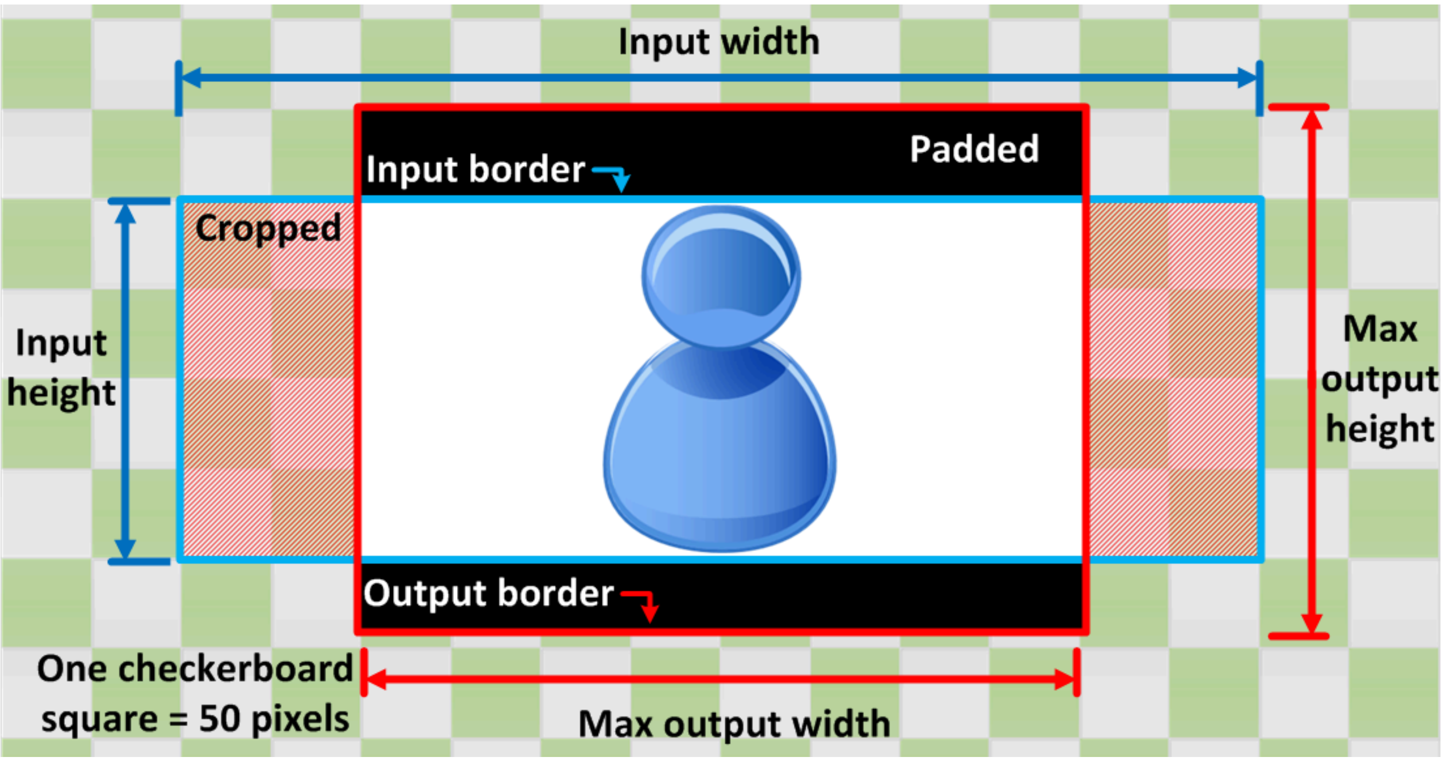
If you choose **Fit without upscaling** for your **Scaling behavior**, MediaConvert decreases the size of your input image until it fits inside the dimensions of your output resolution, without going over any of the dimensions of your output resolution. If your input image is smaller than your output image, MediaConvert does not increase the size of your image.

For example, if your input image is `400` pixels by `400` pixels and you want an output resolution that is `200` pixels by `300` pixels, MediaConvert shrinks your input image to `200` pixels by `200` pixels. MediaConvert does not add padding when you choose **Fit without upscaling**.

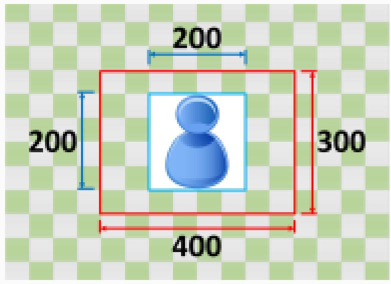
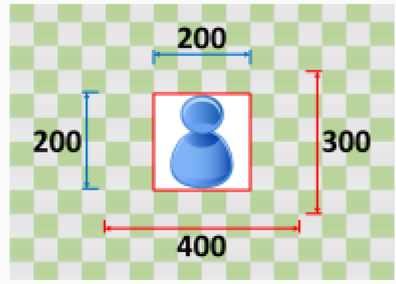
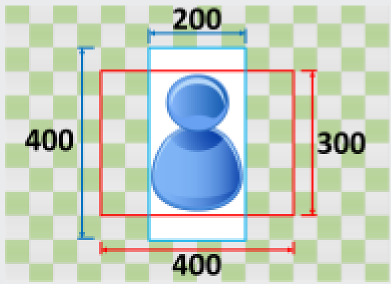
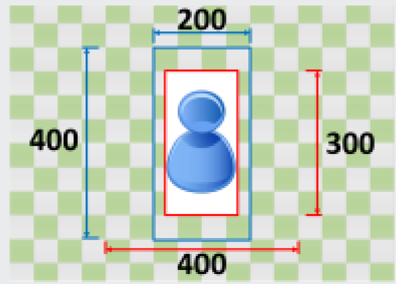
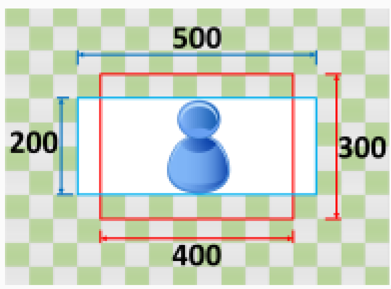
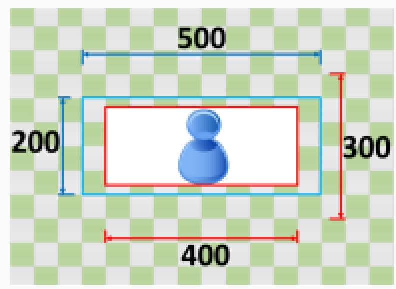
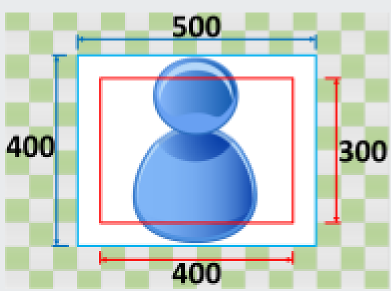
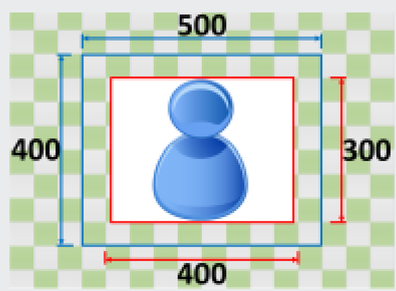
> **Note**
>
> You cannot choose **Fit without upscaling** when you enable **Automated ABR** in **Apple HLS**, **DASH**, or **CMAF** output groups.

Key

The following key graphic shows input and output image width, height, cropping, and padding. In this key graphic, input dimensions (on the top and left) are in blue and output dimensions (on the right and bottom) are in red.



The following table shows example image scaling behavior when your input and output resolutions differ. For details about to read the images, reference the previous key graphic.

Condition	Input	Output
Input width less than output width Input height less than output height		
Input width less than output width Input height greater than output height		
Input width greater than output width Input height less than output height		
Input width greater than output width Input height greater than output height		

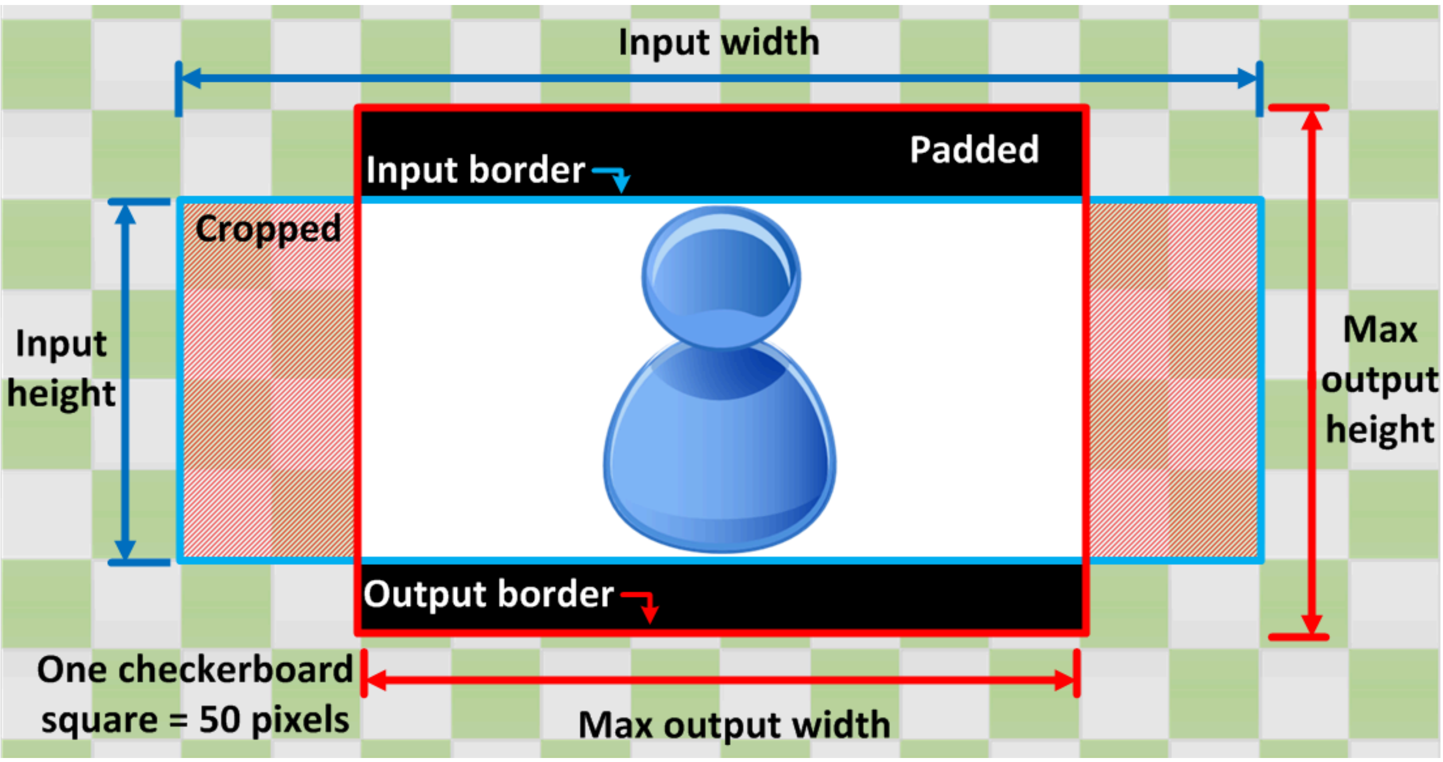
Fill

If you choose **Fill** for your **Scaling behavior**, MediaConvert scales your input image until it fills the dimensions of your output resolution, and crops anything that exceeds the dimensions of your output resolution.

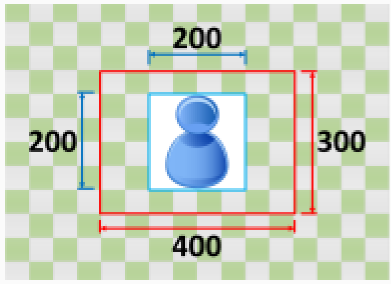
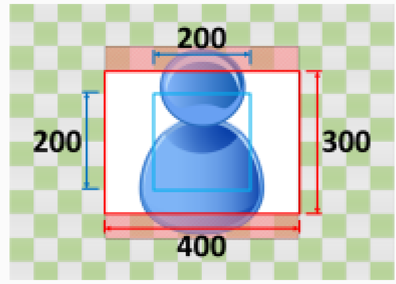
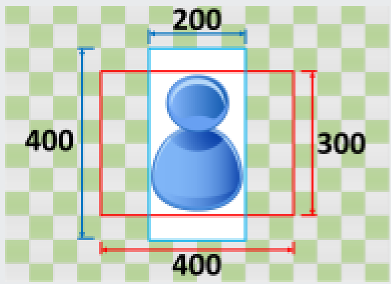
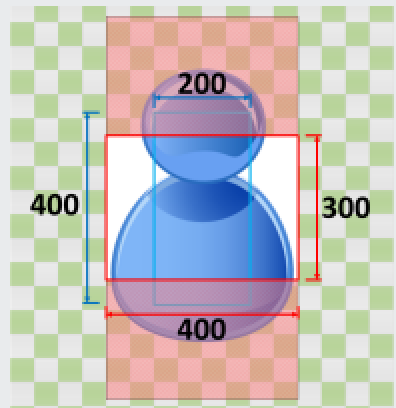
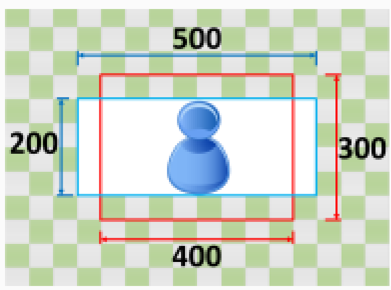
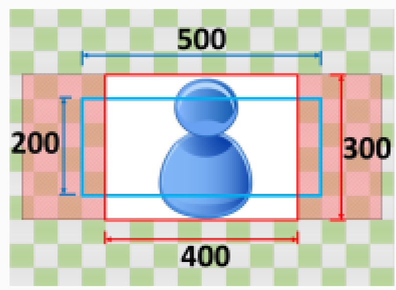
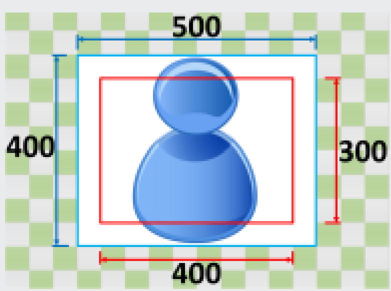
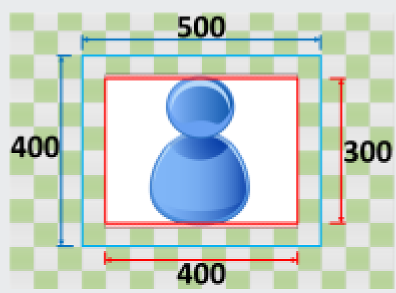
For example, if your input file is 200 pixels by 200 pixels and you want an output resolution that is 300 pixels by 400 pixels, MediaConvert increases the size of your input image to 400 pixels by 400 pixels, crops off the top and bottom 50 pixels, and returns a 300 pixel by 400 pixel file. MediaConvert does not add padding to your output when you choose **Fill**.

Key

The following key graphic shows input and output image width, height, cropping, and padding. In this key graphic, input dimensions (on the top and left) are in blue and output dimensions (on the right and bottom) are in red.



The following table shows example image scaling behavior when your input and output resolutions differ. For details about to read the images, reference the previous key graphic.

Condition	Input	Output
Input width less than output width Input height less than output height		
Input width less than output width Input height greater than output height		
Input width greater than output width Input height less than output height		
Input width greater than output width Input height greater than output height		

Troubleshooting

This chapter describes errors and warnings that you can use to troubleshoot AWS Elemental MediaConvert.

Errors occur when your job has runs into a problem that results in an `ERROR` status. Alternatively, you receive warnings when an issue occurs in your job but the job still completes. For more information about job statuses, see Monitoring MediaConvert job progress.

Errors and warnings consist of both a code and a message. The code provides a unique identifier for the issue, and the message provides a brief note about what happened. For more details, including possible causes and recommended next steps, find your error or warning in the following sections.

If you contact the AWS Support Center for troubleshooting purposes, have the following information available:

- The AWS Region that you used for the job
- The job ID
- Your AWS account ID
- Your source content, if applicable and available
- Any other details about the problem that might assist with troubleshooting

Topics

Error codes

MediaConvert returns error codes when transcoding jobs run into problems. You can use Amazon EventBridge to track the error codes that the service returns.

This table provides detailed information about error codes and messages that MediaConvert returns, their possible causes, and solutions.

> **Note**
>
> MediaConvert charges your account only when a job reaches the `COMPLETED` status. You don't pay for jobs that end in an `ERROR` status.

Error code	Message	Details
1010	Input Error	The service can't open one or more of your input files. Possible causes are as follows: • The file is corrupted. • There's a problem with your AWS Identity and Access Management (IAM) permissions. Check the following: The permissions of the IAM role that the service assumes. The IAM permissions of the person logged in to the service who creates the job. • The input file type isn't supported. See Supported input formats. • The specified path to your Amazon S3 input file has a mistake. To verify the correct file path, choose the **Browse** button on the console and select your

Error code	Message	Details
		file from your Amazon S3 bucket.
1020	Video Error	The service can't find any video in your input stream. The service was able to open and read your input file, but couldn't find a video elementary stream. Possible causes are as follows: • There's a problem with the input file. • The wrong input file was specified.
1021	Audio Error	The service can't find any audio in your input stream. The service was able to open and read your input file, but couldn't find an audio elementary stream. Possible causes are as follows: • There's a problem with the input file. • The wrong input file was specified.
1030	Unsupported Codec	The service doesn't support the codec or container of the input file. See Supported input formats.

Error code	Message	Details
1040	Settings Error	One or more of the job's encoding settings aren't supported in the combination specified. Or, the encoding settings aren't compatible with the input.
1041	Acceleration Settings Error	Your job settings aren't compatible with accelerated transcoding. See Job limitations for accelerated transcoding in AWS Elemental MediaConvert.
1042	Job Doesn't Require Enough Processing Power for Accelerated Transcoding	This job doesn't require enough processing power to benefit from accelerated transcoding. Consider using accelerated transcoding for jobs that would otherwise take 10 minutes or longer to run.

Error code	Message	Details
1056	File Open Error	The service can't open an input or output file. Possible causes are as follows: • A file is corrupted. • There's a problem with your IAM permissions. Check the following: The permissions of the IAM role that the service assumes. The IAM permissions of the person logged in to the service who creates the job. • The input or output file type isn't supported. See Supported input formats. • The specified path to your Amazon S3 input file has a mistake. To verify the correct file path, choose the **Browse** button on the console and select your file from your Amazon S3 bucket.

Error code	Message	Details
1060	Clipping Error	The start and end timecodes specified for an input clip don't exist in the associated input stream. A possible cause is that the input clipping timecodes are specified starting from zero, but the embedded timecodes start at a time other than `00:00:00:00` . To solve this, set both the input **Timecode source** and the **Timecode configuration Source** under **Job settings** to **Start at 0**.
1075	Demuxer Parse Error	MediaConvert couldn't recover from a problematic file when demuxing. Verify that you have provided a supported input. If MediaConvert supports your input, open a case with AWS Support.
1076	Source Read Error	The transcoder couldn't read from one of the input files. The file might have an unexpected end of file. Verify that your input file is valid and that it's truncated properly.

Error code	Message	Details
1080	MXF Output Error	There is a problem in the way that the MXF output settings are configured in your job. Check that these settings are correct and valid per specifica tion.
1091	Encryption Error	There was an error encryptin g one or more of the job outputs. Verify that the IAM role that you chose in your MediaConvert job settings provides access to invoke Amazon API Gateway. For more information about setting up this IAM role, see Set Up IAM Permissions in this guide.
1092	Decode Audio Initialization Error	MediaConvert can't decode one or more of your input audio streams. MediaConv ert supports the codec and container, but there is a problem with the audio source that prevents the transcoder from reading it.

Error code	Message	Details
1401	Permissions Issue	Amazon S3 denied access to a file or bucket. Check the bucket policies on your input and output locations. Also check that the job has the right IAM role specified and that the role has the necessary permissions. For more information, see the following: • Using bucket policies and user policies in the *Amazon S3 User Guide* • Set Up IAM Permissions in this guide

Error code	Message	Details
1404	File Open Error	MediaConvert could not find a file or Amazon S3 bucket. Check that the Amazon S3 bucket and input files exist and that they are specified correctly in the job. Also check that MediaConvert has permissions to access them. For more information, see the following: • Working with Amazon S3 buckets in the *Amazon S3 Developer Guide* • Using bucket policies and user policies in the *Amazon S3 User Guide* • Set Up IAM Permissions in this guide
1432	IAM Role Error	The role specified in your MediaConvert job settings doesn't have the necessary permissions or has another problem. Check that the job has the right IAM role specified and that the role has the correct permissions. For more information about setting up this IAM role, see Set Up IAM Permissions in this guide.

Error code	Message	Details
1433	IAM Role Error	The role specified in your MediaConvert job settings doesn't exist. Check that the job has the right IAM role specified. For more information about setting up this IAM role, see Set Up IAM Permissions in this guide.
1434	IAM Role Error	The role specified in your MediaConvert job settings doesn't have the necessary permissions. Check that the job has the right IAM role specified and that the role has MediaConvert as a trusted entity. For more information about setting up this IAM role, see Set Up IAM Permissions in this guide.

Error code	Message	Details
1515	IAM Role Error	The role specified in your MediaConvert job settings doesn't have the necessary permissions or is malformed . Check that the job has the right IAM role specified and that the role has the correct permissions. For more information about setting up this IAM role, see Set Up IAM Permissions in this guide.
1522	Terminated	There is an unexpected timeout with this job. Contact AWS Support.
1550	Acceleration Fault	There is an unexpected error with the accelerated transcodi ng of this job. Contact AWS Support.
1999	Unknown Error	There is an unexpected transcoding error. Contact AWS Support.
3400	HLS Input Doesn't Meet Requirements	Your HLS input doesn't conform to the requireme nts for MediaConvert supported HLS inputs. For more information, see Using HLS inputs with MediaConv ert.

Error code	Message	Details
3401	HTTP File Access Not Authorized	You specified an HTTP(S) URL for an input file that requires authentication. MediaConvert doesn't pass authentication credentials to the HTTP server. Either change the permissions for your file on the HTTP server, or upload your file to Amazon S3 and specify the Amazon S3 location instead. For more information, see HTTP input requirements.
3403	HTTP Access Forbidden	You specified an HTTP URL for an input file, but the HTTP server refuses the request. Check that you have specified the correct URL. If you have, contact the team that is responsible for maintaining the HTTP server that hosts your file.
3404	HTTP File Not Found	You specified an HTTP URL for an input file, but the HTTP server doesn't have the file. Check that you have specified the correct URL.

Error code	Message	Details
3408	HTTP Upload Error	You specified an HTTP URL for an input file, but the upload failed for a reason that is unrelated to errors 3401, 3403, and 3404. Contact AWS Support.
3450	HTTP Server Error	You specified an HTTP URL for an input file, but the HTTP server returned an error or failed. Contact the team that is responsible for maintaining the HTTP server that hosts your file.
3451	HTTP Server Connection Error	You specified an HTTP URL for an input file, but MediaConvert couldn't connect to the HTTP server. Check that you have specified the correct URL.
3457	Input Disallowed by Policy	You specified an input location that your policy disallows. Specify an allowed input location and resubmit your job.

Error code	Message	Details
3999	HTTP Unknown Error	There was an unexpected error related to retrieving your input file from an HTTP server. Check that you can download your file manually from your HTTP host. If you can, contact AWS Support. If you can't, contact the team that is responsible for maintaining the HTTP server that hosts your file.

Warning codes

AWS Elemental MediaConvert returns warning codes when transcoding jobs run into problems that do not prevent the job from completing. You can use Amazon EventBridge to track the warning codes that the service returns. For more information, see Using EventBridge with AWS Elemental MediaConvert.

This table provides detailed information about the warning codes that MediaConvert returns, including possible causes and solutions.

Warning code	Message	Details
100000	Your job couldn't hop from its original submission queue to its destination queue.	Your job waited in its original submission queue for longer than the wait time that you specified, but it couldn’t move to its new destination queue. Check whether your destination queue still exists. No action is required, but your job may take longer than you expect to complete.

Warning code	Message	Details
		You control hop behavior, including `Wait minutes` and the `Destination queue` , under the **Job management Queue hopping** settings. For more information, see `HopDestination` in the API Reference.
220000	Your input file is truncated.	Your input file is missing data and your output's duration may be shorter than you expect. To troubleshoot, check your input for missing content.

Warning code	Message	Details
230001	Your input's color metadata is missing or incomplete.	MediaConvert could not follow your input's color space because your input has missing or incomplete color metadata. Color metadata includes color primaries, transfer function, and matrix coefficients. Your output might have missing or inaccurate color metadata which can cause players to inaccurately display the video contents. If you specified `Color space conversion` in the output `Color corrector`, note that MediaConvert could not convert the color space and may have written inaccurate color metadata. To resolve, manually specify your input's `Color space` and set `Color space usage` to `Force`. For more information, see `ColorSpace` in the API Reference.

Warning code	Message	Details
230002	MediaConvert can't apply audio duration correction to your input.	There is something wrong with the `'stts'` time-to-sample table in your input file container's audio track, and MediaConvert can't apply audio duration correction. For more information about audio duration correction, see the API Reference. Check your output for any audio video synchronization issues.
230004	Your input is missing information in its `'mdhd'` media header atom.	The `'mdhd'` media header atom in your input is incomplete or is missing data. MediaConvert expects the `'mdhd'` atom to be 32 bytes or 20 bytes. MediaConvert may not read your input correctly. Check the accuracy and quality of your output, including the total file duration and any language codes.

Warning code	Message	Details
230005	MediaConvert can't find color sample range metadata in your input.	MediaConvert could not follow your input's color sample range because your input has missing or incomplete color sample range metadata. Your output might have missing or inaccurate color sample range metadata which can cause players to inaccurately display the video contents. If you specified `Color space conversion` or `Sample range conversion` in the output `Color corrector`, note that your output's sample range may be inaccurate. To resolve, manually specify your input's `Sample range`. For more information, see `SampleRange` in the API Reference.

Warning code	Message	Details
230006	MediaConvert can’t decode a portion of your input audio.	There's something wrong with your input's file structure or audio stream. Check your input for corruptio n or other audio encoding issues. Your output may be missing audio content that MediaConv ert could't decode.
240000	MediaConvert added at least 100 milliseconds of audio silence to keep audio and video in sync.	Your input audio track has missing, corrupt, or unexpecte d data. Check your output for any audio video synchroni zation issues.
240001	MediaConvert dropped at least 100 milliseconds of audio to align audio and video.	Check your output for any audio video synchronization issues.
250001	Your input captions have an unsupported font.	You submitted an input caption with an unsupported font. MediaConvert will use a generic font instead.

Warning code	Message	Details
250002	Your Dolby CBI input has an unsupported bitrate.	You submitted a Dolby CBI input with an unsupported bitrate. MediaConvert will increase it to a supported bitrate. Check that the application generating your DOLBY CBI input is current and up to date.
270000	MediaConvert received a 503 Slow Down error code from Amazon S3 while writing your output to your destination bucket.	While MediaConvert was writing your output files to your destination bucket, it was throttled by Amazon S3. Your job might stall or take longer than expected to complete. You receive a `503 Slow Down` error response when you exceed your request rate limit to Amazon S3. Check for any other applications that are making simultaneous requests and limit them. For more information, see [Troubleshooting Amazon S3]().

Monitoring AWS Elemental MediaConvert

You can track a job's progress, including the status, phase, and percentage of completion. For information, see Monitoring MediaConvert job progress.

You can monitor the *status* of your jobs and their percent completion in these places:

AWS Elemental MediaConvert jobs page

The MediaConvert console shows the following details about your jobs: **Job ID**, **Queue**, **Status**, **Submit time**, **Start Time**, **Transcode duration**, **Job percent complete**, **Finish time**, **Warnings**, **Inputs**, **First input file name**, **First input file path**, **Output groups**, and **First output group destination**.

Optionally, you can filter jobs according to **Status** or **Queue**.

Amazon EventBridge

With EventBridge, you can get more detailed information about the status of your jobs, including any API calls, percent completion of the probing, transcoding, and uploading phases. AWS Elemental MediaConvert sends these events to the default event bus in the EventBridge service. You can create rules to monitor MediaConvert programmatically using other AWS services. For more information, see Using Amazon EventBridge with MediaConvert.

MediaConvert emits metrics, events, and warnings data (when applicable) when a job completes. You can use this data to evaluate the success of a job, even if it completes without error. Data that can be used to evaluate a job, or even a trend in multiple jobs, can be found in these places:

MediaConvert Jobs page

In the MediaConvert console, you can see your job completion status and view any warnings that may have occurred while transcoding. For more granular information, such as QVBR video quality statistics, instead use EventBridge.

MediaConvert API

Through the MediaConvert API, you can request a list of jobs, job status, and job details. For more information, see Jobs in the *AWS Elemental MediaConvert API Reference*.

Amazon EventBridge

With EventBridge, you can get more detailed information about the result of your completed jobs. MediaConvert provides this data through events that it sends to EventBridge. You can subscribe to these events to receive job notifications programmatically or through Amazon SNS, and take actions based on them.

For more information about using events, see Using EventBridge with AWS Elemental MediaConvert.

Amazon CloudWatch metrics

With Amazon CloudWatch, you can view trends in transcoding statistics across your completed jobs, including QVBR video quality. MediaConvert emits transcode statistics to CloudWatch at the end of every job.

For more information about using MediaConvert with CloudWatch metrics, see Using CloudWatch with MediaConvert.

AWS CloudTrail

With CloudTrail, you can get detailed information about when you perform a MediaConvert action through an API call or the MediaConvert console.

For more information about using CloudTrail, see Logging AWS Elemental MediaConvert API calls using AWS CloudTrail.

Topics

Using EventBridge with AWS Elemental MediaConvert

You can use Amazon EventBridge to monitor your AWS Elemental MediaConvert jobs. Here are some examples of what you can do with EventBridge:

Get details about your job outputs

AWS Elemental MediaConvert provides details about your job outputs in the notification for the `COMPLETE` event. This information includes the location and file names of the job's media files and manifests. For details, see Events with COMPLETE status.

For information about job metrics sent to Amazon CloudWatch, see Using CloudWatch with MediaConvert.

Set up email notifications for job status changes

To set up an EventBridge event rule, see Tutorial: Setting up email notifications for failed jobs.

For a list of all job status change notifications that you can set up event rules for, see List of MediaConvert EventBridge events.

Monitor the progress of your job

`STATUS_UPDATE` events provide information about what phase your job is in (`PROBING`, `TRANSCODING`, and `UPLOADING`). For some jobs, MediaConvert provides an estimate of how far your job has progressed. This estimate is shown as a percentage of the total time from when your job leaves its queue to when your output files appear in your output Amazon S3 bucket.

For more information about `STATUS_UPDATE` events, see the table of event types in List of MediaConvert EventBridge events.

For information about adjusting the frequency of status updates, see Adjust the status update interval.

Automatically initiate post-processing with an AWS Lambda function

You can set up EventBridge so that an AWS Lambda function initiates your post-processing code after your job finishes. For more information about using AWS Lambda with AWS Elemental MediaConvert, see one of these resources:

- For experienced cloud architects, see the Video on demand on AWS post on the *AWS Answers* blog.
- For developers new to MediaConvert and Lambda, see the Automating MediaConvert jobs with Lambda tutorial on GitHub.

Get details about jobs you create or other MediaConvert operations

By default, MediaConvert does not emit an EventBridge event for new jobs that you create or any other MediaConvert operation that you perform. To receive EventBridge events in these cases, you first must create an AWS CloudTrail trail.

For more information, see Working with CloudTrail trails.

> **Note**
> EventBridge delivers each event from the MediaConvert event stream at least once. MediaConvert does not require any additional permissions to deliver events to EventBridge.

Topics

Setting up EventBridge rules

To set up Amazon EventBridge rules: Create a rule that links the event from AWS Elemental MediaConvert and the target service that responds to your job status change, such as Amazon Simple Notification Service (Amazon SNS) or AWS Lambda.

For a tutorial on setting up an EventBridge rule with AWS Elemental MediaConvert, see Tutorial: Setting up email notifications for failed jobs.

For a list of the events that MediaConvert sends in the EventBridge event stream, see List of MediaConvert EventBridge events.

For more general information about using EventBridge, see the Amazon EventBridge User Guide.

For troubleshooting information, see Troubleshooting Amazon EventBridge in the *Amazon EventBridge User Guide*.

Topics

Tutorial: Setting up email notifications for failed jobs

In this tutorial, you configure an EventBridge event rule that captures events when a job status changes to `ERROR` and then notifies you about the event. To do this, you first create a topic in Amazon SNS that will send you an email notification about the failed job. Next, you create a rule in EventBridge by defining an event source and referencing the Amazon SNS topic (the "target").

Step 1: Create a topic in Amazon SNS

The first part of setting up an EventBridge rule is preparing the rule target. In this case, that means creating and subscribing to an Amazon SNS topic.

To create an Amazon SNS topic

1. Open the Amazon SNS console at https://console.aws.amazon.com/sns/v3/home.
2. In the navigation pane, choose **Topics**, and then choose **Create topic**.
3. For **Type**, choose **Standard**.
4. For **Name**, enter **`MediaConvertJobErrorAlert`**, and then choose **Create topic**.
5. Choose the topic **Name** for the topic that you just created to view the **Topic details**.
6. On the **MediaConvertJobErrorAlert Topic details** page, in the **Subscriptions** section, choose **Create subscription**.
7. For **Protocol**, choose **Email**. For **Endpoint**, enter the email address that you want Amazon SNS to send the notification to.
8. Choose **Create subscription**.
9. You will receive a notification email from Amazon SNS. When you receive it, choose the **Confirm subscription** link in the email.

Step 2: Specify an event pattern in an EventBridge rule

This step shows how to specify your event pattern in an EventBridge rule. This rule will capture events sent by MediaConvert when a job status changes to `ERROR`.

To set up an event pattern in an EventBridge rule

1. Open the Amazon EventBridge console at https://console.aws.amazon.com/events/.
2. In the navigation pane, choose **Rules**. Keep the **default Event bus** and then choose **Create rule**.
3. For **Name**, enter **`MediaConvertJobStateError`**, and then choose **Next**.
4. In the **Event pattern** section, starting with **Event source** choose the following settings:
   - **Event source**: **`AWS services`**
   - **AWS service**: **`MediaConvert`**
   - **Event type**: **`MediaConvert Job State Change`**
   - **Event type, Specific state(s)**: **`ERROR`**
5. An **Event pattern** box will look like the following example.

```
{
  "source": ["aws.mediaconvert"],
  "detail-type": ["MediaConvert Job State Change"],
  "detail": {
    "status": ["ERROR"]
  }
}
```

   This code defines an EventBridge event rule that matches any event where the job status changes to `ERROR`. For more information about event patterns, see Events and event patterns in the *Amazon CloudWatch User Guide*.
6. Choose **Next**.

Step 3: Add the Amazon SNS topic and finish your rule

Add the Amazon SNS topic that you created in step 1 to the EventBridge rule that you started in step 2.

To add the Amazon SNS topic and finish the EventBridge rule

1. In the **Select target(s)** section, under **Select a target**, choose **SNS topic**.
2. For **Topic**, choose **MediaConvertJobErrorAlert**.
3. Choose **Next**.

4. Optionally add tags. Then choose **Next**.
5. Review your settings. Then choose **Create rule**.

Step 4: Test your rule

To test your rule, submit a job that you know will cause an error. For example, specify an input location that does not exist. If you configured your event rule correctly, you should receive an email with the event text message in a few minutes.

To test the rule

1. Open the AWS Elemental MediaConvert console at https://console.aws.amazon.com/mediaconvert.
2. Submit a new MediaConvert job. For more information, see [Configuring jobs in MediaConvert](#).
3. Check the email account that you specified when you set up your Amazon SNS topic. Confirm that you received an email notification for the job error.

Monitoring MediaConvert job progress

MediaConvert provides status information for each job that you create. You can monitor a job's status to determine what is happening with your job, track its progress, or toubleshoot issues.

Topics

Job statuses

Each job you submit will progress through a number of different possible statuses. Successful jobs end with a `COMPLETE` status, and unsuccessful jobs end with an `ERROR` or `CANCELED` status.

Successful jobs

A successful job results in all of its outputs being written to your Amazon S3 bucket. The following list contains details about the different statuses successful jobs go through:

1. SUBMITTED: Jobs that you created in the MediaConvert console or through a CreateJob operation that MediaConvert hasn't started processing.
2. INPUT_INFORMATION: MediaConvert read details about your input or inputs.
3. PROGRESSING: MediaConvert began processing your job.
4. STATUS_UPDATE: MediaConvert has been processing your job for at least a minute or longer. Jobs with this status go through three different *job phases*. For more information, see the section called "Job phases".
5. COMPLETE: MediaConvert completed your job and all of your outputs were saved to your Amazon S3 bucket.

Successful jobs may also include NEW_WARNING or QUEUE_HOP statuses.

Unsuccessful jobs

An unsuccessful job results in no outputs being written to your Amazon S3 bucket, although it may initially progress similar to a successful job. Unsuccessful jobs have one of the the following two statuses:

- ERROR: MediaConvert had an issue processing your job. For more information, see the section called "Error codes".
- CANCELED: You canceled a job in the MediaConvert console or through a CancelJob operation.

How to find a job's status

To view a list of all job statuses in the current Region, open the Jobs page in the MediaConvert console. Or submit a GetJob or ListJobs operation.

AWS CloudTrail records details about each create job request you make. CloudTrail only emits an EventBridge event for SUBMITTED jobs if you also create a CloudTrail trail. For more information, see Working with CloudTrail trails and Accessing AWS service events via AWS CloudTrail.

MediaConvert emits an EventBridge event when a job begins processing and then whenever its status changes, for example if it changes from PROGRESSING to COMPLETE or ERROR. For list of EventBridge events, see the section called "List of MediaConvert EventBridge events".

Job phases

If a job's status is `PROGRESSING` for over a minute, MediaConvert emits a `STATUS_UPDATE` event with the current job *phase*. A job phase provides details about what is happening with a job. Job phases progress through the following order:

1. `PROBING`: When your job is in the `PROBING` phase, MediaConvert reads the information from your input files that the service needs for transcoding.
2. `TRANSCODING`: When your job is in the `TRANSCODING` phase, the service demuxes, decodes, encodes, and remuxes your content. In some jobs, the service begins uploading outputs to your output Amazon S3 bucket during this phase. The phase ends when all transcoding is complete.
3. `UPLOADING`: When your job is in the `UPLOADING` phase, the service uploads your transcoded outputs to your output Amazon S3 bucket. In the case of outputs that the service begins to upload during the `TRANSCODING` phase, the `UPLOADING` phase begins when the transcoding is done. It continues until all uploads are finished.

To view a job's phase, open the Jobs page in the MediaConvert console, check the STATUS_UPDATE EventBridge event, or submit a `GetJob` or `ListJobs` operation.

List of MediaConvert EventBridge events

AWS Elemental MediaConvert emits an event to Amazon EventBridge when the status of a job changes. You can create EventBridge rules for any of these events.

Most jobs will only emit a few of these events, with the most common being: `INPUT_INFORMATION`, `PROGRESSING`, and `COMPLETE`.

For more information about each event type, choose the link in the **Event** column.

MediaConvert events

Event	Sent when	Contains
INPUT_INFORMATION	Soon after MediaConvert begins processing the job.	Media information, such as frame height and width, frame rate, and codec.

Event	Sent when	Contains
		Information from MediaConv ert about all inputs in a single event.
PROGRESSING	A job moves from the `SUBMITTED` state to the `PROGRESSING` state.	Basic job details.
STATUS_UPDATE	Approximately one minute after MediaConvert begins processing the job. Sent approximately every minute after that, until the job is completed or encounters an error.	Job progress expressed in the number of frames transcode d since the beginning of the job.
COMPLETE	A job is completed and MediaConvert writes all outputs successfully without errors.	Warnings and output information about the completed job.
CANCELED	A job is canceled.	Basic job details.
ERROR	A job has an error. At least one output has an error.	The error code or codes and any messages. Includes any other ephemeral job information about the job's error status.
NEW_WARNING	A warning condition arises.	The warning code or codes and any warning messages.
QUEUE_HOP	When a job hops queues.	The ARNs for both queues and the job's priority within the queue.

> **Note**
>
> MediaConvert does not emit a SUBMITTED event. To receive an EventBridge event any time you make an API call, including calls from the MediaConvert console, you must create a AWS CloudTrail trail. For more information, see Accessing AWS service events via AWS CloudTrail and Working with CloudTrail trails.

When MediaConvert sends an event to EventBridge, the following fields are present in the resulting JSON.

- **version** — Currently 0 (zero) for all events.
- **id** — A Version 4 UUID generated for every event.
- **detail-type** — The type of event that's being sent.
- **source** — Identifies the service that generated the event.
- **account** — The 12-digit AWS account ID that ran the job.
- **time** — The time the event occurred.
- **region** — Identifies the AWS Region of the job.
- **resources** — A JSON array that contains the Amazon Resource Name (ARN) of the job.
- **detail** — A JSON object that contains information about the job.

The following sections contain event message details, JSON responses, and event patterns for every EventBridge event that MediaConvert emits.

Topics

Events with INPUT_INFORMATION status

MediaConvert sends the event for `INPUT_INFORMATION` after a job begins processing and after MediaConvert reads information about your input. The event primarily contains media information about your input, such as frame height and width, frame rate, and codec.

MediaConvert includes information about all of your inputs in a single event.

The following JSON is an example event containing the `INPUT_INFORMATION` status for a job with a single input.


```
{
    "version": "0",
    "id": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "detail-type": "MediaConvert Job State Change",
    "source": "aws.mediaconvert",
    "account": "111122223333",
    "time": "2022-12-19T19:07:12Z",
    "region": "us-west-2",
    "resources": [
        "arn:aws:mediaconvert:us-west-2:111122223333:jobs/1671476818694-phptj0"
    ],
    "detail": {
        "timestamp": 1671476832075,
        "accountId": "111122223333",
        "queue": "arn:aws:mediaconvert:us-west-2:111122223333:queues/Default",
        "jobId": "1671476818694-phptj0",
        "status": "INPUT_INFORMATION",
        "userMetadata": {},
        "inputDetails": [
            {
                "id": 1,
                "uri": "s3://DOC-EXAMPLE-BUCKET/file/file.mp4",
                "audio": [
                    {
                        "channels": 2,
                        "codec": "AAC",
                        "language": "UND",
                        "sampleRate": 44100,
                        "streamId": 2
                    }
                ],
                "video": [
```

```
                    {
                        "bitDepth": 8,
                        "codec": "H_264",
                        "colorFormat": "YUV_420",
                        "fourCC": "avc1",
                        "frameRate": 24,
                        "height": 1080,
                        "interlaceMode": "PROGRESSIVE",
                        "sar": "1:1",
                        "standard": "UNSPECIFIED",
                        "streamId": 1,
                        "width": 1920
                    }
                ]
            }
        ]
    }
}
```

You can use the following sample JSON to create an EventBridge event pattern for jobs with a status of `INPUT_INFORMATION`.

```
{
  "source": ["aws.mediaconvert"],
  "detail-type": ["MediaConvert Job State Change"],
  "detail": {
    "status": ["INPUT_INFORMATION"]
  }
}
```

Events with PROGRESSING status

MediaConvert sends the event for `PROGRESSING` when a job moves from the `SUBMITTED` state to the `PROGRESSING` state.

> **Note**
>
> After a job has been in the `PROGRESSING` status for 48 hours, the service puts it into an `ERROR` state and stops working on it. You are not billed for jobs that end in an `ERROR` state.

The following JSON is an example event containing the `PROGRESSING` status for a job.

```
{
    "version": "0",
    "id": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "detail-type": "MediaConvert Job State Change",
    "source": "aws.mediaconvert",
    "account": "111122223333",
    "time": "2022-12-19T19:20:21Z",
    "region": "us-west-2",
    "resources": [
        "arn:aws:mediaconvert:us-west-2:111122223333:jobs/1671477617078-2886ye"
    ],
    "detail": {
        "timestamp": 1671477621654,
        "accountId": "111122223333",
        "queue": "arn:aws:mediaconvert:us-west-2:111122223333:queues/Default",
        "jobId": "1671477617078-2886ye",
        "status": "PROGRESSING",
        "userMetadata": {}
    }
}
```

You can use the following sample JSON to create an EventBridge event pattern for jobs with a status of PROGRESSING.

```
{
    "source": ["aws.mediaconvert"],
    "detail-type": ["MediaConvert Job State Change"],
    "detail": {
        "status": ["PROGRESSING"]
    }
}
```

Events with STATUS_UPDATE status

MediaConvert sends the event for STATUS_UPDATE approximately one minute after it begins processing a job. By default, updates are sent approximately every minute after that, until the service finishes transcoding or encounters an error.

You can optionally specify a different update frequency in your job, with the **Status update interval** setting. For more frequent updates, you can choose 10, 12, 15, 20, or 30 seconds. For fewer updates, you can choose anywhere from 2 to 10 minutes, in increments of one minute. For more information, see Adjust the status update interval.

A `STATUS_UPDATE` event contains information about your job's current phase. When available, it contains job and phase percent completion.

Job phases are as follows:

- During `PROBING`, the service reads information about the input while it prepares to transcode.
- During `TRANSCODING`, the service demuxes, decodes, encodes, and remuxes your content. In some jobs, the service begins uploading outputs to your output Amazon S3 bucket during this phase, as well. The phase ends when all transcoding is complete.
- During `UPLOADING`, the service uploads the remaining transcoded outputs to your Amazon S3 bucket.

The following JSON is an example event containing the `STATUS_UPDATE` status for a job.


```
{
    "version": "0",
    "id": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "detail-type": "MediaConvert Job State Change",
    "source": "aws.mediaconvert",
    "account": "111122223333",
    "time": "2022-12-19T19:21:21Z",
    "region": "us-west-2",
    "resources": [
        "arn:aws:mediaconvert:us-west-2:111122223333:jobs/1671477617078-2886ye"
    ],
    "detail": {
        "timestamp": 1671477681737,
        "accountId": "111122223333",
        "queue": "arn:aws:mediaconvert:us-west-2:111122223333:queues/Default",
        "jobId": "1671477617078-2886ye",
        "status": "STATUS_UPDATE",
        "userMetadata": {},
        "framesDecoded": 353,
        "jobProgress": {
            "phaseProgress": {
                "PROBING": {
                    "status": "COMPLETE",
                    "percentComplete": 100
```

```
                },
                "TRANSCODING": {
                    "status": "PROGRESSING",
                    "percentComplete": 2
                },
                "UPLOADING": {
                    "status": "PENDING",
                    "percentComplete": 0
                }
            },
            "jobPercentComplete": 7,
            "currentPhase": "TRANSCODING",
            "retryCount": 0
        }
    }
}
```

You can use the following sample JSON to create an EventBridge event pattern for jobs with a status of STATUS_UPDATE.

```
{
    "source": ["aws.mediaconvert"],
    "detail-type": ["MediaConvert Job State Change"],
    "detail": {
        "status": ["STATUS_UPDATE"]
    }
}
```

Adjust the status update interval

By default, AWS Elemental MediaConvert sends STATUS_UPDATE events to Amazon EventBridge approximately once per minute. These status updates provide information about how your job is progressing. You can adjust the status update interval by specifying a different update frequency in your job.

To specify the STATUS_UPDATE frequency

1. On the **Create job** page, in the **Job** pane on the left, in the **Job settings** section, choose **AWS integration**.
2. In the **AWS integration** section on the right, for **Status update interval (sec)**, choose **interval, in seconds, between updates**.

If you use the API or an SDK, you can find this setting in the JSON file of your job. The setting name is statusUpdateInterval.

Events with COMPLETE status

MediaConvert sends the event for `COMPLETE` when all outputs are written to Amazon S3 without errors. It contains both warnings and output information for the completed job. For more information about output file names and paths, see Output file names and paths.

The following JSON is an example event containing the `COMPLETE` status for a job.

```
{
    "version": "0",
    "id": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "detail-type": "MediaConvert Job State Change",
    "source": "aws.mediaconvert",
    "account": "111122223333",
    "time": "2022-12-19T19:07:12Z",
    "region": "us-west-2",
    "resources": [
        "arn:aws:mediaconvert:us-west-2::jobs/1671476818694-phptj0"
    ],
    "detail": {
        "timestamp": 1671476832124,
        "accountId": "111122223333",
        "queue": "arn:aws:mediaconvert:us-west-2:111122223333:queues/Default",
        "jobId": "1671476818694-phptj0",
        "status": "COMPLETE",
        "userMetadata": {},
        "warnings": [
            {
                "code": 000000,
                "count": 1
            }
        ],
        "outputGroupDetails": [
            {
                "outputDetails": [
                    {
                        "outputFilePaths": [
                            "s3://DOC-EXAMPLE-BUCKET/file/file.mp4"
                        ],
                        "durationInMs": 30041,
```

```
                        "videoDetails": {
                            "widthInPx": 1920,
                            "heightInPx": 1080,
                            "qvbrAvgQuality": 7.38,
                            "qvbrMinQuality": 7,
                            "qvbrMaxQuality": 8,
                            "qvbrMinQualityLocation": 2168,
                            "qvbrMaxQualityLocation": 25025
                        }
                    }
                ],
                "type": "FILE_GROUP"
            }
        ],
        "paddingInserted": 0,
        "blackVideoDetected": 10,
        "blackSegments": [
            {
                "start": 0,
                "end": 10
            }
        ]
    }
}
```


COMPLETE events contain additional information about your job and outputs. The following table lists and describes the different properties available in job event message details.

COMPLETE event message details

Property	Data type	Details
paddingInserted	integer	The total duration of blank frames MediaConvert inserted across all outputs in your job, in milliseconds. Video padding inserts blank frames to help keep audio and video durations aligned. Large paddingInserted values show that more blank

Property	Data type	Details
		frames were inserted. These values also show to what extend your input audio tracks start late, or end early, or both.
`qvbrAvgQuality`	float	The average video quality of your Quality-Defined Variable Bitrate (QVBR) output. Included for QVBR outputs only.
`qvbrMinQuality`	float	The minimum video quality detected in your QVBR output. Included for QVBR outputs only.
`qvbrMaxQuality`	float	The maximum video quality detected in your QVBR output. Included for QVBR outputs only.

Property	Data type	Details
qvbrMinQualityLocation	integer	The location in your output where qvbrMinQuality was detected, in milliseconds. You can use qvbrMinQualityLocation while reviewing your output video quality and bandwidth usage. Included for QVBR outputs only.
qvbrMaxQualityLocation	integer	The location in your output where qvbrMaxQuality was detected, in milliseconds. You can use qvbrMaxQualityLocation while reviewing your output video quality and bandwidth usage. Included for QVBR outputs only.
warnings code count	array integer integer	Any warning codes seen in the job and the number of times they occurred. For more information, see Warning codes.

Property	Data type	Details
`blackVideoDetected`	integer	The total duration of black video frames in your outputs that are also present in your inputs, in milliseconds. `blackVideoDetected` does not include any black frames inserted by MediaConvert.

`blackVideoSegments`
 	
`start`
 	
`end`
	array
	
	integer
	
	integer
		The location or locations in your output where black video frames were detected.
		
		Each segment of black video in your output is shown with its own start and end.
		
		`blackVideoSegments` does not include any black frames inserted by MediaConvert.

`averageBitrate`	integer	The average bitrate of your video output, calculated by dividing the duration by the total bits.

You can use the following sample JSON to create an EventBridge event pattern for jobs with a status of `COMPLETE`.

```
{
  "source": ["aws.mediaconvert"],
  "detail-type": ["MediaConvert Job State Change"],
  "detail": {
    "status": ["COMPLETE"]
```

```
    }
}
```

Output file names and paths

The EventBridge job `COMPLETE` notification includes details about your output in JSON. This information includes the file names and paths for the outputs of the job—including manifests and media assets.

The files that AWS Elemental MediaConvert creates depends on the output groups that you set up in the job. For example, DASH ISO packages contain an .mpd manifest and .mp4 media fragment files.

You can find output file name and path information in the following properties:

`playlistFilePaths`

A list of the Amazon S3 file paths to the top-level manifests.

`outputFilePaths`

The file path to either the media or the manifest, depending on the output group type.

`type`

The type of output group, which determines what files are listed in the `playlistFilePaths` and `outputFilePaths`.

The following table summarizes the values for these properties, depending on the output group type.

Type	playlistFilePaths	outputFilePaths
`FILE_GROUP` (standard output)	*not returned*	File name and path of the media file. Example: `s3://DOC-EXAMPLE-BUCKET/file/file.mp4`

Type	playlistFilePaths	outputFilePaths
`FILE_GROUP` (with additional frame capture output)	*not returned*	File name and path of the final captured image. Example: `s3://DOC-EXAMPLE-BUCKET/frameoutput/file.0000036.jpg`
`HLS_GROUP`	File name and path of the top-level manifest. Example: `s3://DOC-EXAMPLE-BUCKET/hls/main.m3u8`	File name and path of the manifests for the individual outputs. Examples: • `s3://DOC-EXAMPLE-BUCKET/hls/mainv1.m3u8` • `s3://DOC-EXAMPLE-BUCKET/hls/mainv2.m3u8`
`DASH_ISO_GROUP`	File name and path of the manifest. Example: `s3://DOC-EXAMPLE-BUCKET/dash/1.mpd`	*not returned*

Type	playlistFilePaths	outputFilePaths
`CMAF_GROUP`	File name and path for each of the top-level manifests. Examples: • `s3://DOC-EXAMPLE-BUCKET/cmaf/1.mpd` • `s3://DOC-EXAMPLE-BUCKET/cmaf/1.m3u8`	*not returned*
`MS_SMOOTH_GROUP`	File name and path of the server-side manifest. Example: `s3://DOC-EXAMPLE-BUCKET/smooth/1.ism`	File name and path of the video manifests for each of the individual outputs. Examples: • `s3://DOC-EXAMPLE-BUCKET/smooth/1_va.ismv` • `s3://DOC-EXAMPLE-BUCKET/smooth/2_va.ismv`

For sample responses in JSON for each output group type, see the following topics:

Topics

- File group
- File group with a frame capture output
- Apple HLS group
- DASH ISO group
- CMAF group
- Microsoft Smooth Streaming group

File group

The following is an Amazon EventBridge sample event for a job with a `COMPLETE` status. The example includes output file path information for a file group.


```
{
    "detail": {
        "status": "COMPLETE",
        "paddingInserted": 0,
        "blackVideoDetected": 10,
        "blackSegments": [
            "start": 0,
            "end": 10
        ],
        "outputGroupDetails": [
            {
                "outputDetails": [
                    {
                        "outputFilePaths": [
                            "s3://DOC-EXAMPLE-BUCKET/file/file.mp4"
                        ],
                        "durationInMs": 180041,
                        "videoDetails": {
                            "averageBitrate": 200000,
                            "widthInPx": 1280,
                            "heightInPx": 720,
                            "qvbrAvgQuality": 7.38,
                            "qvbrMinQuality": 7,
                            "qvbrMaxQuality": 8,
                            "qvbrMinQualityLocation": 2168,
                            "qvbrMaxQualityLocation": 25025
                        },
                    }
                ],
                "type": "FILE_GROUP"
            }
        ],
        "timestamp": 1536964380391,
        "accountId": "111122223333",
        "queue": "arn:aws:mediaconvert:us-west-2:111122223333:queues/Default",
        "jobId": "1536964333549-opn151",
        "userMetadata": {},
        "warnings": [
            {
```

```
                "code": 000000,
                "count": 1
            }
        ],
    },
    "version": "0",
    "id": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "detail-type": "MediaConvert Job State Change",
    "source": "aws.mediaconvert",
    "account": "111122223333",
    "time": "2018-09-14T21:54:31Z",
    "region": "us-west-2",
    "resources": [
        "arn:aws:mediaconvert:us-west-2:111122223333:jobs/1536961999428-kxngbl"
    ]
}
```


> **Note**
>
> QVBR statistics are only available when your video output uses Quality-Defined Variable Bitrate (QVBR) rate control.

File group with a frame capture output

This guide provides an Amazon EventBridge example event for a job with a `COMPLETE` status. It also shows output file path information for a file group with a frame capture output.

A *frame capture output* is an output that you set up to create still images of video. You set it up similar to a regular **File group** output group. However, you remove the audio component, choose **No container** for the container, and then choose **Frame capture to JPEG** for the video codec.

> **Note**
>
> You can create frame capture outputs only in jobs that also have a regular audio and video output. MediaConvert doesn't support jobs that consist only of a frame capture output.

When you create a frame capture output, the `COMPLETE` status includes the `outputFilePaths` property. This tells you the file name and path of the final captured image.

> **Tip**
>
> Because the service includes automatic numbering in the frame capture file names, you can infer all the image names from the final one. For example, if your `outputFilePaths` value is `s3://DOC-EXAMPLE-BUCKET/frameoutput/file.0000036.jpg`, you can infer that there are 35 other images in the same location, named `file.0000001`, `file.0000002`, and so on.

The following is an EventBridge sample event for a job with a `COMPLETE` status. It includes output file path information for a file group with a frame capture output.


```
{
    "detail": {
        "status": "COMPLETE",
        "paddingInserted": 0,
        "blackVideoDetected": 10,
        "blackSegments": [
                {
                    "start": 0,
                    "end": 10
                }
        ],
        "outputGroupDetails": [
            {
                "outputDetails": [
                    {
                        "outputFilePaths": [
                            "s3://DOC-EXAMPLE-BUCKET/frameoutput/file.0000036.jpg"
                        ],
                        "durationInMs": 185000,
                        "videoDetails": {
                            "widthInPx": 1280,
                            "heightInPx": 720
                        }
                    }
                ],
                "type": "FILE_GROUP"
            },
            {
                "outputDetails": [
                    {
```

```
                        "outputFilePaths": [
                            "s3://DOC-EXAMPLE-BUCKET/file/file.mp4"
                        ],
                        "durationInMs": 180041,
                        "blackVideoDurationInMs": 0,
                        "videoDetails": {
                            "widthInPx": 1280,
                            "heightInPx": 720,
                            "averageQVBRScore": 7.38,
                            "minimumQVBRScore": 7,
                            "maximumQVBRScore": 8,
                            "minimumQVBRScoreLocationInMs": 2168,
                            "maximumQVBRScoreLocationInMs": 25025
                        }
                    }
                ],
                "type": "FILE_GROUP"
            }
        ],
        "timestamp": 1536964380391,
        "accountId": "111122223333",
        "queue": "arn:aws:mediaconvert:us-west-2:111122223333:queues/Default",
        "jobId": "1536964333549-opn151",
        "userMetadata": {},
        "warnings": [
            {
                "code": 000000,
                "count": 1
            }
        ]
    },
    "version": "0",
    "id": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "detail-type": "MediaConvert Job State Change",
    "source": "aws.mediaconvert",
    "account": "111122223333",
    "time": "2018-09-14T21:54:31Z",
    "region": "us-west-2",
    "resources": [
        "arn:aws:mediaconvert:us-west-2:111122223333:jobs/1536961999428-kxngbl"
    ]
}
```

> **Note**
>
> Quality-Defined Variable Bitrate (QVBR) statistics are only available when your video output uses QVBR rate control.

Apple HLS group

The following is an Amazon EventBridge sample event for a job with a `COMPLETE` status. It includes output file path information for an Apple HLS group.


```
{
    "detail": {
        "status": "COMPLETE",
        "paddingInserted": 0,
        "blackVideoDetected": 10,
        "blackSegments": [
            {
                "start": 0,
                "end": 10
            }
        ],
        "outputGroupDetails": [
            {
                "outputDetails": [
                    {
                        "outputFilePaths": [
                            "s3://DOC-EXAMPLE-BUCKET/hls/mainv2.m3u8"
                        ],
                        "durationInMs": 180041,
                        "blackVideoDurationInMs": 0,
                        "videoDetails": {
                            "widthInPx": 426,
                            "heightInPx": 240,
                            "averageQVBRScore": 7.38,
                            "minimumQVBRScore": 7,
                            "maximumQVBRScore": 8,
                            "minimumQVBRScoreLocationInMs": 2168,
                            "maximumQVBRScoreLocationInMs": 25025
                        }
                    },
                    {
```

```
                        "outputFilePaths": [
                            "s3://DOC-EXAMPLE-BUCKET/hls/mainv1.m3u8"
                        ],
                        "durationInMs": 180041,
                        "blackVideoDurationInMs": 0,
                        "videoDetails": {
                            "widthInPx": 1280,
                            "heightInPx": 720,
                            "averageQVBRScore": 7.38,
                            "minimumQVBRScore": 7,
                            "maximumQVBRScore": 8,
                            "minimumQVBRScoreLocationInMs": 2168,
                            "maximumQVBRScoreLocationInMs": 25025
                        }
                    }
                ],
                "type": "HLS_GROUP",
                "playlistFilePaths": [
                    "s3://DOC-EXAMPLE-BUCKET/hls/main.m3u8"
                ]
            }
        ],
        "timestamp": 1536964380391,
        "accountId": "111122223333",
        "queue": "arn:aws:mediaconvert:us-west-2:111122223333:queues/Default",
        "jobId": "1536964333549-opn151",
        "userMetadata": {},
        "warnings": [
            {
                "code": 000000,
                "count": 1
            }
        ]
    },
    "version": "0",
    "id": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "detail-type": "MediaConvert Job State Change",
    "source": "aws.mediaconvert",
    "account": "111122223333",
    "time": "2018-09-14T21:54:31Z",
    "region": "us-west-2",
    "resources": [
        "arn:aws:mediaconvert:us-west-2:111122223333:jobs/1536961999428-kxngbl"
    ]
```

```
}
```

Note

Quality-Defined Variable Bitrate (QVBR) statistics are only available when your video output uses QVBR rate control.

DASH ISO group

The following is an Amazon EventBridge sample event for a job with a `COMPLETE` status. It includes output file path information for a DASH ISO group.


```
{
    "detail": {
        "status": "COMPLETE",
        "paddingInserted": 0,
        "blackVideoDetected": 10,
        "blackSegments": [
            {
                "start": 0,
                "end": 10
            }
        ],
        "outputGroupDetails": [
            {
                "outputDetails": [
                    {
                        "durationInMs": 180041,
                        "blackVideoDurationInMs": 0,
                        "videoDetails": {
                            "widthInPx": 1280,
                            "heightInPx": 720,
                            "averageQVBRScore": 7.38,
                            "minimumQVBRScore": 7,
                            "maximumQVBRScore": 8,
                            "minimumQVBRScoreLocationInMs": 2168,
                            "maximumQVBRScoreLocationInMs": 25025
                        }
                    }
                ],
                "type": "DASH_ISO_GROUP",
```

```
                "playlistFilePaths": [
                    "s3://DOC-EXAMPLE-BUCKET/dash/1.mpd"
                ]
            }
        ],
        "timestamp": 1536964380391,
        "accountId": "111122223333",
        "queue": "arn:aws:mediaconvert:us-west-2:111122223333:queues/Default",
        "jobId": "1536964333549-opn151",
        "userMetadata": {},
        "warnings": [
            {
                "code": 000000,
                "count": 1
            }
        ]
    },
    "version": "0",
    "id": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "detail-type": "MediaConvert Job State Change",
    "source": "aws.mediaconvert",
    "account": "111122223333",
    "time": "2018-09-14T21:54:31Z",
    "region": "us-west-2",
    "resources": [
        "arn:aws:mediaconvert:us-west-2:111122223333:jobs/1536961999428-kxngbl"
    ]
}
```

Note

Quality-Defined Variable Bitrate (QVBR) statistics are only available when your video output uses QVBR rate control.

CMAF group

The following is an EventBridge sample event for a job with a `COMPLETE` status. It includes output file path information for a CMAF group.

```
{
    "detail": {
```

```
        "status": "COMPLETE",
        "paddingInserted": 0,
        "blackVideoDetected": 10,
        "blackSegments": [
            {
                "start": 0,
                "end": 10
            }
        ],
        "outputGroupDetails": [
            {
                "outputDetails": [
                    {
                            "durationInMs": 180041,
                            "blackVideoDurationInMs": 0,
                            "videoDetails": {
                                "widthInPx": 1280,
                                "heightInPx": 720,
                                "averageQVBRScore": 7.38,
                                "minimumQVBRScore": 7,
                                "maximumQVBRScore": 8,
                                "minimumQVBRScoreLocationInMs": 2168,
                                "maximumQVBRScoreLocationInMs": 25025
                            }
                    }
                ],
                "type": "CMAF_GROUP",
                "playlistFilePaths": [
                    "s3://DOC-EXAMPLE-BUCKET/cmaf/1.mpd",
                    "s3://DOC-EXAMPLE-BUCKET/cmaf/1.m3u8"
                ]
            }
        ],
        "timestamp": 1536964380391,
        "accountId": "111122223333",
        "queue": "arn:aws:mediaconvert:us-west-2:111122223333:queues/Default",
        "jobId": "1536964333549-opn151",
        "userMetadata": {},
        "warnings": [
            {
                "code": 000000,
                "count": 1
            }
        ]
```

```
    },
    "version": "0",
    "id": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "detail-type": "MediaConvert Job State Change",
    "source": "aws.mediaconvert",
    "account": "111122223333",
    "time": "2018-09-14T21:54:31Z",
    "region": "us-west-2",
    "resources": [
        "arn:aws:mediaconvert:us-west-2:111122223333:jobs/1536961999428-kxngbl"
    ]
}
```


Note

Quality-Defined Variable Bitrate (QVBR) statistics are only available when your video output uses QVBR rate control.

Microsoft Smooth Streaming group

The following is an EventBridge sample event for a job with a `COMPLETE` status. It includes output file path information for a Microsoft Smooth Streaming group.


```
{
    "detail": {
        "status": "COMPLETE",
        "paddingInserted": 0,
        "blackVideoDetected": 10,
        "blackSegments": [
            {
                "start": 0,
                "end": 10
            }
        ],
        "outputGroupDetails": [
            {
                "outputDetails": [
                    {
                        "outputFilePaths": [
                            "s3://DOC-EXAMPLE-BUCKET/smooth/1_va.ismv"
                        ],
```

```
                        "durationInMs": 180041,
                        "blackVideoDurationInMs": 0,
                        "videoDetails": {
                            "widthInPx": 1280,
                            "heightInPx": 534,
                            "averageQVBRScore": 7.38,
                            "minimumQVBRScore": 7,
                            "maximumQVBRScore": 8,
                            "minimumQVBRScoreLocationInMs": 2168,
                            "maximumQVBRScoreLocationInMs": 25025
                        }
                    }
                ],
                "type": "MS_SMOOTH_GROUP",
                "playlistFilePaths": [
                    "s3://DOC-EXAMPLE-BUCKET/smooth/1.ism"
                ]
            }
        ],
        "timestamp": 1536964380391,
        "accountId": "111122223333",
        "queue": "arn:aws:mediaconvert:us-west-2:111122223333:queues/Default",
        "jobId": "1536964333549-opn151",
        "userMetadata": {},
        "warnings": [
            {
                "code": 000000,
                "count": 1
            }
        ]
    },
    "version": "0",
    "id": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "detail-type": "MediaConvert Job State Change",
    "source": "aws.mediaconvert",
    "account": "111122223333",
    "time": "2018-09-14T21:54:31Z",
    "region": "us-west-2",
    "resources": [
        "arn:aws:mediaconvert:us-west-2:111122223333:jobs/1536961999428-kxngbl"
    ]
}
```

> **Note**
>
> Quality-Defined Variable Bitrate (QVBR) statistics are only available when your video output uses QVBR rate control.

Events with CANCELED status

MediaConvert sends the event for `CANCELED` when the job is canceled. It contains basic job details including `timestamp`, `accountID`, `queue`, `jobId`, and `userMetadata`.

The following JSON is an example event containing the `CANCELED` status for a job.

```
{
    "version": "0",
    "id": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "detail-type": "MediaConvert Job State Change",
    "source": "aws.mediaconvert",
    "account": "111122223333",
    "time": "2018-01-07T23:35:20Z",
    "region": "us-west-2",
    "resources": ["arn:aws:mediaconvert:us-west-2:111122223333:jobs/1515368087458-
qnoxtd"],
    "detail": {
        "timestamp": 1515368120764,
        "accountId": "111122223333",
        "queue": "arn:aws:mediaconvert:us-west-2:111122223333:queues/Default",
        "jobId": "1515368087458-qnoxtd",
        "status": "CANCELED",
        "userMetadata": {}
    }
}
```

You can use the following sample JSON to create an EventBridge event pattern for jobs with a status of `CANCELED`.

```
{
    "source": ["aws.mediaconvert"],
    "detail-type": ["MediaConvert Job State Change"],
    "detail": {
        "status": ["CANCELED"]
    }
```

```
}
```

Events with ERROR status

MediaConvert sends the event for ERROR when at least one output has an error. It contains the error code or codes, messages, and warnings or other ephemeral job information about the job's error status. For more information about error codes, see Error codes.

The following JSON is an example event containing the ERROR status for a job.


```
{
    "version": "0",
    "id": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "detail-type": "MediaConvert Job State Change",
    "source": "aws.mediaconvert",
    "account": "111122223333",
    "time": "2018-01-07T23:35:20Z",
    "region": "us-west-2",
    "resources": ["arn:aws:mediaconvert:us-west-2:111122223333:jobs/1515368087458-
qnoxtd"],
    "detail": {
        "timestamp": 1515368120764,
        "accountId": "111122223333",
        "queue": "arn:aws:mediaconvert:us-west-2:111122223333:queues/Default",
        "jobId": "1515368087458-qnoxtd",
        "status": "ERROR",
        "errorCode": 1040,
        "errorMessage": "Example error message",
        "userMetadata": {}
    }
}
```


You can use the following sample JSON to create an EventBridge event pattern for jobs with a status of ERROR.

```
{
    "source": ["aws.mediaconvert"],
    "detail-type": ["MediaConvert Job State Change"],
    "detail": {
        "status": ["ERROR"]
    }
}
```

Events with NEW_WARNING status

MediaConvert sends an event for `NEW_WARNING` when a warning condition arises. A warning condition doesn't stop the job from running. It contains the submission queue ARN, the job ID, and a warning message. Warning messages inform you about conditions that don't stop the job but might indicate that the job is not progressing as you planned.

For example, a job that you set up to hop queues reaches the wait time that you specified, but can't hop.

`NEW_WARNING` only contains the most recent warning message. It doesn't report previous warnings, even if those warning conditions are still in effect.

For more information about warning messages, see Warning codes.

The following JSON is an example event containing the `NEW_WARNING` status for a job.

```
{
    "version": "0",
    "id": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "detail-type": "MediaConvert Job State Change",
    "source": "aws.mediaconvert",
    "account": "111122223333",
    "time": "2018-01-07T23:35:20Z",
    "region": "us-west-2",
    "resources": ["arn:aws:mediaconvert:us-west-2:111122223333:jobs/1515368087458-
qnoxtd"],
    "detail": {
        "timestamp": 1515368120764,
        "accountId": "111122223333",
        "queue": "arn:aws:mediaconvert:us-west-2:111122223333:queues/Default",
        "jobId": "1515368087458-qnoxtd",
        "status": "NEW_WARNING",
        "warningCode": "000000",
        "warningMessage": "Example warning message",
        "userMetadata": {}
    }
}
```

You can use the following sample JSON to create an EventBridge event pattern for jobs with a status of `NEW_WARNING`.

```
{
  "source": ["aws.mediaconvert"],
  "detail-type": ["MediaConvert Job State Change"],
  "detail": {
    "status": ["NEW_WARNING"]
  }
}
```


Events with QUEUE_HOP status

MediaConvert sends the `QUEUE_HOP` event when a job hops queues. It contains the ARNs for both queues and the job's priority within the queues.

Both `priority` and `previousPriority` will be the same unless the queue hopping configuration specifies a new priority to give to the job as it hops queues.

The following JSON is an example event containing the `QUEUE_HOP` status for a job.


```
{
    "version": "0",
    "id": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "detail-type": "MediaConvert Job State Change",
    "source": "aws.mediaconvert",
    "account": "111122223333",
    "time": "2018-01-07T23:35:20Z",
    "region": "us-west-2",
    "resources": ["arn:aws:mediaconvert:us-west-2:111122223333:jobs/1515368087458-
qnoxtd"],
    "detail": {
        "timestamp": 1515368120764,
        "accountId": "111122223333",
        "queue": "arn:aws:mediaconvert:us-west-2:111122223333:queues/Default",
        "previousQueue": "arn:aws:mediaconvert:us-west-2:111122223333:queues/
Alternate",
        "priority": 0,
        "previousPriority": 0,
        "jobId": "1515368087458-qnoxtd",
        "status": "QUEUE_HOP",
        "userMetadata": {}
    }
}
```

You can use the following sample JSON to create an EventBridge event pattern for jobs with a status of QUEUE_HOP.

```
{
    "source": ["aws.mediaconvert"],
    "detail-type": ["MediaConvert Job State Change"],
    "detail": {
        "status": ["QUEUE_HOP"]
    }
}
```

Using CloudWatch with MediaConvert

You can use Amazon CloudWatch to gather metrics about your AWS Elemental MediaConvert jobs. Here are some examples of what you can do with CloudWatch:

Get details about your requests, queues, jobs, and outputs

Metrics are grouped according to dimension, including Operation metrics, Queue metrics, and Job metrics.

- *Operation metrics* include any errors that occur when you interact with MediaConvert.
- *Queue metrics* include details about all of your jobs in a given queue, like the total number of jobs, or the total duration of your outputs.
- *Job metrics* include details about your outputs that you can use to identify trends, statistics, or potential issues with your video workflow.

For more information, see List of MediaConvert CloudWatch metrics.

Create CloudWatch dashboards and alarms

You can create CloudWatch dashboards to track things for a job queue. For example, you can track transcoding jobs that have completed or that have errors. You can also use metrics such as `StandbyTime` to create alarms to detect if there is a large backlog for a job queue. For more information, see Creating dashboards and alarms for AWS Elemental MediaConvert using Amazon CloudWatch.

Topics

- List of MediaConvert CloudWatch metrics

List of MediaConvert CloudWatch metrics

AWS Elemental MediaConvert sends the following metrics to CloudWatch at the end of every job.

Metric name	Unit	Description
`Errors`	Count	Tracks errors encountered when making a particular API call. Operations include: `CreateJob` , `GetJob`, `ListJobs`, `ListPresets` , `ListQueues` , `ListTagsForResourc e` , and `Subscribe`
`AudioOutputDuration`	Milliseconds	The number of milliseconds of audio-only output for a queue.
`SDOutputDuration`	Milliseconds	The number of milliseconds of standard definition (SD) output for a queue. For the definition of each video resolutio n category, see the AWS Elemental MediaConvert pricing page.
`HDOutputDuration`	Milliseconds	The number of milliseconds of high-defi nition (HD) output for a queue. For the definition of each video resolutio n category, see the AWS Elemental MediaConvert pricing page.
`UHDOutputDuration`	Milliseconds	The number of milliseconds of ultra-high-definition (UHD) output for a queue. For the definition of each video resolutio n category, see the AWS Elemental MediaConvert pricing page.

Metric name	Unit	Description
8KOutputDuration	Milliseconds	The number of milliseconds of 8K output for a queue. For the definition of each video resolution category, see the AWS Elemental MediaConvert pricing page.
JobsCompletedCount	Count	The number of jobs completed in a queue.
JobsCanceled	Count	The number of jobs canceled in a queue.
JobsErroredCount	Count	The number of jobs that failed because of invalid inputs, such as a request to transcode a file that is not in the specified input bucket.
StandbyTime	Milliseconds	The amount of time before MediaConvert starts transcoding a job, in milliseconds.
TranscodingTime	Milliseconds	The number of milliseconds for MediaConvert to complete transcoding.
BlackVideoDetected	Milliseconds	The total duration of black video frames in your output that are also present in your input in milliseconds. BlackVideoDetected does not include any black frames inserted by MediaConvert.
BlackVideoDetectedRatio	Ratio	The ratio of black video frames to the total duration of your output. Higher ratios show outputs with more black frames.

Metric name	Unit	Description
LongestBlackSegmentDetected	Milliseconds	The location in your output with the longest continuous segment of black video frames, in milliseconds.
VideoPaddingInserted	Milliseconds	The total duration of blank frames MediaConvert inserted across all outputs in your job, in milliseconds. Video padding inserts blank frames to help keep audio and video durations aligned. Large VideoPaddingInserted values show that more blank frames were inserted. They also show to what extent your input audio tracks either start late, end early, or both.
VideoPaddingInsertedRatio	Ratio	The ratio of blank frames MediaConvert inserted to the overall duration of your output. Higher ratios may signal audio video sync issues in your input.
AvgBitrateTop	Bits per second	The average bitrate of the highest bitrate output in the output group.
AvgBitrateBottom	Bits per second	The average bitrate of the lowest bitrate output in the output group.
QVBRAvgQualityHighBitrate	Score	The average QVBR quality score of the highest bitrate output in the output group.
QVBRAvgQualityLowBitrate	Score	The average QVBR quality score of the lowest bitrate output in the output group.

Metric name	Unit	Description
QVBRMinQualityHigh Bitrate	Score	The minimum QVBR quality score of the highest bitrate output in the output group.
QVBRMinQualityLowB itrate	Score	The minimum QVBR quality score of the lowest bitrate output in the output group.

Logging AWS Elemental MediaConvert API calls using AWS CloudTrail

AWS Elemental MediaConvert integrates with AWS CloudTrail, a service that provides a record of operations taken by a user, role, or an AWS service. CloudTrail captures all operations (API calls) you perform, including those from the MediaConvert Console, as events. Events contain information about requests to MediaConvert, including the IP address, who made the request, when it was made, the MediaConvert operation, and additional details. For more information about CloudTrail events, see What are CloudTrail events?

A few examples of what CloudTrail can help you find include: when you submitted a create job request, who deleted a queue, or what tags were added to a resource.

For a complete list of all MediaConvert operations you can perform, see the MediaConvert API Reference.

To learn more about CloudTrail, see the AWS CloudTrail User Guide.

Topics

Finding information about MediaConvert in CloudTrail

You can view, search, and download events from the last 90 days in Event history in the CloudTrail console. For more information, see Working with CloudTrail Event history.

For an ongoing record of CloudTrail events beyond the last 90 days, you must create a CloudTrail *trail* or CloudTrail Lake event data store.

CloudTrail trails

With CloudTrail trails, CloudTrail delivers log files for events to an Amazon S3 bucket. Additionally, you can configure other AWS services to further analyze and act upon the event data collected. When you create a trail, CloudTrail also emits each event to Amazon EventBridge. For more information about CloudTrail trails, see Working with CloudTrail trails.

CloudTrail Lake event data stores

With CloudTrail Lake event data stores, CloudTrail provides a searchable store of event data. For more information, see Working with CloudTrail Lake.

Understanding CloudTrail management events for MediaConvert

Management events provide information about management operations that are performed on resources in your AWS account. These are also known as control plane operations, or MediaConvert operations. By default, CloudTrail logs all management events.

CloudTrail management events represent a single request from any source. They include information about where a request originated, who made the request, when the request was made, the MediaConvert operation, and other important details.

For details about the contents of these events, see CloudTrail record contents.

Origin

Events include the source IP address under `sourceIPAddress`, AWS Region under `awsRegion`, and user agent (browser or client info) under `userAgent` to help you find where a request originated.

Identity

Events include identity information, under `userIdentity`, to help you find who made the request. You can use this information to determine the following:

- Whether the request was made with root user or user credentials.
- Whether the request was made on behalf of an IAM Identity Center user.

- 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, including a list of relevant fields, see the CloudTrail userIdentity element.

Time

Events include a time stamp, under `eventTime`.

MediaConvert operation

Events include the MediaConvert operation under `eventName`. For example, `CreateJob` when you submit a new job, or `CreateQueue` when you request a new Queue.

Other important details

Some events include specific details about your MediaConvert request under `requestParameters`, as well as MediaConvert's response under `responseElements`.

For example, if you submit a `CreateJob` request, the `requestParameters` field will include the JSON for the job settings that you submitted. If successful, the event also includes the complete and validated job settings JSON, as well as the job ID, under `responseElements`.

For another example, if you submit a `CreateQueue` request, the `requestParameters` field will include the the queue settings that you submitted. If successful, the event will also include the queue ARN under `responseElements`.

> **Note**
>
> MediaConvert only supports recording management events in CloudTrail, it does not support data events. For details about these event types, and their differences, see CloudTrail concepts.

Example events

The following examples show a CloudTrail event for the `CreateJob`, `CreateQueue`, `DeleteQueue`, and `TagResource` operations. CloudTrail also records all other MediaConvert operations, though they are not shown here.

Example event: CreateJob

```
{
    "eventVersion": "1.09",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAIOSFODNN7EXAMPLE:example-admin",
        "arn": "arn:aws:sts::111122223333:assumed-role/admin/example-admin",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAIOSFODNN7EXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/admin",
                "accountId": "111122223333",
                "userName": "admin"
            },
            "attributes": {
                "creationDate": "2024-04-04T17:30:19Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2024-04-04T17:45:26Z",
    "eventSource": "mediaconvert.amazonaws.com",
    "eventName": "CreateJob",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "203.0.113.100",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36",
    "requestParameters": {
        "settings": {...},
        "accelerationSettings": {
            "mode": "DISABLED"
        },
        "role": "arn:aws:iam::111122223333:role/service-role/
MediaConvert_Default_Role",
        "clientRequestToken": "1712252705233-zyxwvut",
        "statusUpdateInterval": "SECONDS_60",
        "billingTagsSource": "JOB",
        "priority": 0,
        "queue": "arn:aws:mediaconvert:us-west-2:111122223333:queues/Default"
    },
```

```
    "responseElements": {
        "job": {
            "arn": "arn:aws:mediaconvert:us-west-2:111122223333:jobs/1712252725875-
defhgi",
            "id": "1712252725875-defhgi",
            "createdAt": 1712252726,
            "queue": "arn:aws:mediaconvert:us-west-2:111122223333:queues/Default",
            "role": "arn:aws:iam::111122223333:role/service-role/
MediaConvert_Default_Role",
            "settings": {...},
            "status": "SUBMITTED",
            "timing": {
                "submitTime": 1712252726
            },
            "billingTagsSource": "JOB",
            "accelerationSettings": {
                "mode": "DISABLED"
            },
            "statusUpdateInterval": "SECONDS_60",
            "priority": 0,
            "accelerationStatus": "NOT_APPLICABLE",
            "messages": {
                "info": [],
                "warning": []
            },
            "clientRequestToken": "1712252705233-abcDEF"
        }
    },
    "requestID": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "eventID": "0987dcba-09fe-87dc-65ba-ab0987654321",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Example event: CreateQueue

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
```

```
        "principalId": "AKIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/testuser",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "testUser",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-07-10T14:01:57Z"
            }
        },
        "invokedBy": "signin.amazonaws.com"
    },
    "eventTime": "2018-07-10T16:49:13Z",
    "eventSource": "mediaconvert.amazonaws.com",
    "eventName": "CreateQueue",
    "awsRegion": "eu-west-1",
    "sourceIPAddress": "203.0.113.100",
    "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
 like Gecko) Chrome/67.0.3396.99 Safari/537.36",
    "requestParameters": {
        "name": "QueueName",
        "description": "Example queue description.",
        "tags": {}
    },
    "responseElements": {
        "queue": {
            "arn": "arn:aws:mediaconvert:eu-west-1:111122223333:queues/QueueName",
            "createdAt": 1531241353,
            "lastUpdated": 1531241353,
            "type": "CUSTOM",
            "status": "ACTIVE",
            "description": "",
            "name": "QueueName",
            "submittedJobsCount": 0,
            "progressingJobsCount": 0
        }
    },
    "requestID": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "eventID": "0987dcba-09fe-87dc-65ba-ab0987654321",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
```

```
}
```

Example event: DeleteQueue


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AKIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/testuser",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "testuser",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-07-10T14:01:57Z"
            }
        },
        "invokedBy": "signin.amazonaws.com"
    },
    "eventTime": "2018-07-10T15:36:29Z",
    "eventSource": "mediaconvert.amazonaws.com",
    "eventName": "DeleteQueue",
    "awsRegion": "eu-west-1",
    "sourceIPAddress": "203.0.113.100",
    "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
 like Gecko) Chrome/67.0.3396.99 Safari/537.36",
    "requestParameters": {
        "name": "QueueName"
    },
    "responseElements": null,
    "requestID": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "eventID": "0987dcba-09fe-87dc-65ba-ab0987654321",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```


Example event: TagResource


```
{
    "eventVersion": "1.05",
```

```
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AKIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/testuser",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "testuser"
    },
    "eventTime": "2018-07-10T18:44:27Z",
    "eventSource": "mediaconvert.amazonaws.com",
    "eventName": "TagResource",
    "awsRegion": "eu-west-1",
    "sourceIPAddress": "203.0.113.100",
    "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
 like Gecko) Chrome/67.0.3396.99 Safari/537.36",
    "requestParameters": {
        "arn": "arn:aws:mediaconvert:eu-west-1:111122223333:queues/ExampleQueue",
        "Tags": {
            "CostCenter": "Example-Tag"
        }
    },
    "responseElements": null,
    "requestID": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "eventID": "0987dcba-09fe-87dc-65ba-ab0987654321",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

Tagging AWS Elemental MediaConvert resources

A *tag* is a 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 as "test." Tags help you to identify and organize your AWS resources. Tags that you assign to AWS Elemental MediaConvert jobs, job templates, queues, and presets are integrated with tags across AWS services.

Popular use cases for these tags are as follows:

- You can activate these tags on the AWS Billing and Cost Management dashboard, and then set up a monthly cost allocation report. For more information, see Setting up resources for cost allocation through tagging.
- You can allow or deny resource-level access to your resources using AWS Identity and Access Management (IAM). For more information, see Controlling access to AWS resources using resource tags in the *IAM User Guide*.

Topics

Setting up AWS Elemental MediaConvert resources for cost allocation through tagging

For all outputs that you produce using an on-demand queue, you can use the AWS Billing and Cost Management dashboard to set up a monthly cost allocation report. This report shows what AWS charges you for transcoding, sorted by resource. You can set up your jobs so that your job outputs are sorted by tags on the job or on a resource that you use to create the job. That is, you can sort your bill by the tags that you put on the job, on the queue that you submit the job to, on the job template that you create the job from, or on the output presets that you use to set up the individual outputs of the job.

To set up cost allocation through tagging for your AWS Elemental MediaConvert charges

1. Tag the resources that you intend to sort your bill by. For instructions, see the other topics in this chapter.
2. Create your transcoding jobs, specifying how you want your costs allocated as follows:

    a. On the **Create job** page, in the **Job** pane on the left, under **Job settings**, choose **AWS integration**.

    b. In the **Job settings** section on the right, under **AWS integration**, for **Billing tag source**, choose which tags you want to use to sort the job's outputs. You can choose to sort by tags on a resource that you use to create the job—job template, output preset, or queue. Or you can choose **Job** to sort by the tags on the job itself.

    > **Note**
    >
    > Jobs, and the tags on them, persist for only 90 days. If your workflow references tags over longer periods of time, use tags on the queue, job template, or output preset rather than tags on the job.

3. Activate these tags on the AWS Billing and Cost Management dashboard. For more information, see Activating user-defined cost allocation tags in the *Billing and Cost Management User Guide*.
4. Set up your report. For more information, see Monthly cost allocation report in the *Billing and Cost Management User Guide*.

Adding tags when you create an AWS Elemental MediaConvert resource

The following procedures show you how to add tags to your MediaConvert queues, job templates, and output presets when you create them.

Topics

Adding tags when creating a resource (console)

You can add tags when you create a queue, job template, or output preset.

To add tags when you create a queue, job template, or output preset (console)

1. Follow the steps in one of the following procedures to begin creating the resource, but don't save the resource:
    - Configuring jobs in MediaConvert
    - Create an on-demand queue
    - Creating a custom template from scratch
    - Creating a custom preset from scratch
    - Creating a custom preset from a system preset
2. Find the **Tags** section in the relevant location:
    - For jobs – on the **Create job** page, after you choose **Settings** from the **Job** section on the left
    - For queues – at the bottom of the **Create queue** page
    - For output presets – at the bottom of the **Create preset** page
    - For job templates – at the bottom of the **Create job template** page, after you choose **Settings** from the **Job** section on the left
3. In the **Tags** section, choose **Add**.
4. For **Tag key**, enter a name for the tag. For **Tag value**, enter a value for the tag.
5. Choose **Create** to save the new resource with its tags.

Adding tags when creating a resource (API and AWS CLI)

When you create a job, job template, output preset, or queue using the AWS Elemental MediaConvert API or the AWS CLI, submit your JSON specification for the resource as usual. Include tags as shown in the following JSON example, in `tags`:

```
{
 "name": "Job Template Test with Resource Tags",
 "description": "Job Template Test",
 "tags":{
  "Company": "Banana",
  "Stage": "Production"
```

```
},
"settings":{
```

Adding tags to an existing AWS Elemental MediaConvert resource

The following procedure shows you how to add tags to existing job templates, output presets, or queues using the AWS Elemental MediaConvert console. You can't use the console to add tags to existing jobs.

For information about adding tags using the API, see the `POST` method in the Tags endpoint section of the *MediaConvert API Reference*.

> **Note**
>
> Tags on your MediaConvert resources don't appear in the JSON response to a `GET` request on the resource. Instead, send a `GET` request to the `Tags` endpoint. If you send your request directly to the API, rather than using an SDK, you must URL encode the resource ARN.

To add tags to job templates, output presets, and queues (console)

1. Open the MediaConvert console at https://console.aws.amazon.com/mediaconvert.
2. Choose the three-bar icon on the left to access the left navigation pane.
3. Choose **Output presets**, **Job templates**, or **Queues**.
4. Choose the name of the specific resource that you want to add a tag to.
5. Choose the **Update**, **Edit queue**, or **Update preset** button in the upper right.
6. In the **Tags** section at the bottom of the page, choose **Add**.
7. For **Tag key**, enter a name for the tag. For **Tag value**, enter a value for the tag.
8. Choose **Save**.

Viewing tags on an AWS Elemental MediaConvert resource

The following procedure shows you how to view tags on existing queues, job templates, and output presets using the AWS Elemental MediaConvert console.

To do this using the API, see the GET method in the Tags arn endpoint section of the *MediaConvert API Reference*. If you send your request directly to the API, rather than using an SDK, you must URL encode the resource ARN.

To view tags for queues, job templates, and output presets (console)

1. Open the MediaConvert console at https://console.aws.amazon.com/mediaconvert.
2. Choose the three-bar icon on the left to access the left navigation pane.
3. Choose **Job templates**, **Output presets**, or **Queues**.
4. Choose the name of the specific resource that has tags that you want to view.
5. View the tags for the resource in the **Tags** section at the bottom of the page.

Editing tags on an AWS Elemental MediaConvert resource

The following procedure shows you how to edit tags on existing job templates, output presets, and queues using the AWS Elemental MediaConvert console. You can't edit tags on existing jobs.

To do this using the API, see the POST method in the Tags endpoint section of the *MediaConvert API Reference*.

To edit tags for job templates, output presets, and queues (console)

1. Open the MediaConvert console at https://console.aws.amazon.com/mediaconvert.
2. Choose the three-bar icon on the left to access the left navigation pane.
3. Choose **Job templates**, **Output presets**, or **Queues**.
4. Choose the name of the specific resource that has tags that you want to change.
5. Choose the **Update**, **Edit queue**, or **Update preset** button in the upper right.
6. In the **Tags** section at the bottom of the page, edit any values for **Tag key** and **Tag values** that you want to change.
7. Choose **Save**.

Removing tags from an AWS Elemental MediaConvert resource

The following procedure shows you how to remove tags from existing job templates, output presets, and queues using the AWS Elemental MediaConvert console.

To do this using the API, see the PUT method in the [Tags] endpoint section of the *MediaConvert API Reference*.

To remove tags from a job template, output preset, or queue (console)

1. Open the MediaConvert console at https://console.aws.amazon.com/mediaconvert.
2. Choose the three-bar icon on the left to access the left navigation pane.
3. Choose **Job templates**, **Output presets**, or **Queues**.
4. Choose the name of the specific resource that has tags that you want to change.
5. Choose the **Update**, **Edit queue**, or **Update preset** button in the upper right.
6. Next to any tag that you want to delete, choose **Remove**.
7. Choose **Save**.

Restrictions for tags on AWS Elemental MediaConvert resources

The following basic restrictions apply to tags:

- Maximum number of tags per resource – 50.
- Maximum **Key** length – 128 Unicode characters.
- Maximum **Value** length – 256 Unicode characters.
- Valid values for **Key** and **Value** – Uppercase and lowercase letters in the UTF-8 character set, numbers, space, and the following characters: _ . : / = + - and @.
- Tag keys and values are case sensitive.
- Don't use the `aws:` prefix for either keys or values. It's reserved for AWS use.

Using metadata tags with AWS Elemental MediaConvert jobs

Unless you have existing integrations or workflows that rely on metadata (`userMetadata`) tags, we recommend that you use standard AWS tags for both automatic integration with AWS services and for custom integrations and workflows. For more information, see Tagging AWS Elemental MediaConvert resources.

A *tag* is a 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 as "test." Tags help you to identify and organize your AWS resources. Metadata tags that you assign to AWS Elemental MediaConvert jobs appear in Amazon EventBridge notifications.

You add metadata tags to jobs in one of the following ways:

- Through the MediaConvert console on the **Job settings** page, in the **Metadata** pane.
- Through the MediaConvert API in your job settings JSON payload. Include tags as shown in the following JSON snippet, in `userMetadata`. For more information, see the Jobs endpoint section of the *MediaConvert API Reference*.

```
{
 "name": "Job Template Test with Resource Tags",
 "description": "Job Template Test",
 "userMetadata":{
  "Company": "Banana",
  "Stage": "Production"
 },
 "settings":{
```

AWS Elemental MediaConvert job tag restrictions

The following basic restrictions apply to tags on jobs:

- Maximum number of tags per job – 10.
- Maximum **Key** length – 128 Unicode characters.
- Maximum **Value** length – 256 Unicode characters.
- Valid values for **Key** and **Value** – Uppercase and lowercase Unicode letters in any language, numbers, space, and the following characters: _ . : / = + - and @.
- Tag keys and values are case sensitive.
- Don't use the `aws:` prefix for either keys or values. It's reserved for AWS use.

Security in AWS Elemental MediaConvert

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 MediaConvert, see AWS Services in scope by compliance program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using MediaConvert. The following topics show you how to configure MediaConvert to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your MediaConvert resources.

Topics

General AWS data protection

The AWS shared responsibility model applies to data protection in AWS Elemental MediaConvert. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-2.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with MediaConvert or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

Identity and Access Management for AWS Elemental MediaConvert

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 MediaConvert resources. IAM is an AWS service that you can use with no additional charge.

Topics

Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in MediaConvert.

Service user – If you use the MediaConvert service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more MediaConvert features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator. If you cannot access a feature in MediaConvert, see Troubleshooting AWS Elemental MediaConvert identity and access.

Service administrator – If you're in charge of MediaConvert resources at your company, you probably have full access to MediaConvert. It's your job to determine which MediaConvert features and resources your service users should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with MediaConvert, see How AWS Elemental MediaConvert works with IAM.

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to MediaConvert. To view example MediaConvert identity-based policies that you can use in IAM, see Identity-based policy examples for AWS Elemental MediaConvert.

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. You must be *authenticated* (signed in to AWS) as the AWS account root user, as an IAM user, or by assuming an IAM role.

You can sign in to AWS as a federated identity by using credentials provided through an identity source. AWS IAM Identity Center (IAM Identity Center) users, your company's single sign-on authentication, and your Google or Facebook credentials are examples of federated identities. When you sign in as a federated identity, your administrator previously set up identity federation using IAM roles. When you access AWS by using federation, you are indirectly assuming a role.

Depending on the type of user you are, you can sign in to the AWS Management Console or the AWS access portal. For more information about signing in to AWS, see How to sign in to your AWS account in the *AWS Sign-In User Guide*.

If you access AWS programmatically, AWS provides a software development kit (SDK) and a command line interface (CLI) to cryptographically sign your requests by using your credentials. If you don't use AWS tools, you must sign requests yourself. For more information about using the recommended method to sign requests yourself, see Signing AWS API requests in the *IAM User Guide*.

Regardless of the authentication method that you use, you might be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see Multi-factor authentication in the *AWS IAM Identity Center User Guide* and Using multi-factor authentication (MFA) in AWS in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you don't use the root user for your everyday tasks. Safeguard your root user credentials and use them to perform the tasks that only the root user can perform. For the complete list of tasks that require you to sign in as the root user, see Tasks that require root user credentials in the *IAM User Guide*.

Federated identity

As a best practice, require human users, including users that require administrator access, to use federation with an identity provider to access AWS services by using temporary credentials.

A *federated identity* is a user from your enterprise user directory, a web identity provider, the AWS Directory Service, the Identity Center directory, or any user that accesses AWS services by using credentials provided through an identity source. When federated identities access AWS accounts, they assume roles, and the roles provide temporary credentials.

For centralized access management, we recommend that you use AWS IAM Identity Center. You can create users and groups in IAM Identity Center, or you can connect and synchronize to a set of users and groups in your own identity source for use across all your AWS accounts and applications. For information about IAM Identity Center, see What is IAM Identity Center? in the *AWS IAM Identity Center User Guide*.

IAM users and groups

An *IAM user* is an identity within your AWS account that has specific permissions for a single person or application. Where possible, we recommend relying on temporary credentials instead of creating IAM users who have long-term credentials such as passwords and access keys. However, if you have specific use cases that require long-term credentials with IAM users, we recommend that you rotate access keys. For more information, see Rotate access keys regularly for use cases that require long-term credentials in the *IAM User Guide*.

An *IAM group* is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named *IAMAdmins* and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see When to create an IAM user (instead of a role) in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity within your AWS account that has specific permissions. It is similar to an IAM user, but is not associated with a specific person. You can temporarily assume an IAM role in the AWS Management Console by switching roles. You can assume a role by calling an AWS CLI or

AWS API operation or by using a custom URL. For more information about methods for using roles, see Using IAM roles in the *IAM User Guide*.

IAM roles with temporary credentials are useful in the following situations:

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see Creating a role for a third-party Identity Provider in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permissions sets, see Permission sets in the *AWS IAM Identity Center User Guide*.
- **Temporary IAM user permissions** – An IAM user or role can assume an IAM role to temporarily take on different permissions for a specific task.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
  - **Forward access sessions (FAS)** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.
  - **Service role** – A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.

  - **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.
- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Using an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

To learn whether to use IAM roles or IAM users, see When to create an IAM role (instead of a user) in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when a principal (user, root user, or role session) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

By default, users and roles have no permissions. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Creating IAM policies in the *IAM User Guide*.

Identity-based policies can be further categorized as *inline policies* or *managed policies*. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, see Choosing between managed policies and inline policies in the *IAM User Guide*.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Access control lists (ACLs)

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs. To learn more about ACLs, see Access control list (ACL) overview in the *Amazon Simple Storage Service Developer Guide*.

Other policy types

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries –** A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of an entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs) –** SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see How SCPs work in the *AWS Organizations User Guide*.
- **Session policies –** Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see Session policies in the *IAM User Guide*.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the *IAM User Guide*.

How AWS Elemental MediaConvert works with IAM

Before you use IAM to manage access to MediaConvert, learn what IAM features are available to use with MediaConvert.

IAM features you can use with AWS Elemental MediaConvert

IAM feature	MediaConvert support
Identity-based policies	Yes
Resource-based policies	No
Policy actions	Yes
Policy resources	Yes
Policy condition keys (service-specific)	Yes
ACLs	No
ABAC (tags in policies)	Partial
Temporary credentials	Yes
Forward access sessions (FAS)	Yes
Service roles	Yes
Service-linked roles	No

To get a high-level view of how MediaConvert and other AWS services work with most IAM features, see AWS services that work with IAM in the *IAM User Guide*.

Identity-based policies for MediaConvert

Supports identity-based policies: Yes

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Creating IAM policies in the *IAM User Guide*.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. You can't specify the principal in an identity-based policy because it applies to the user or role to which it is attached. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

Identity-based policy examples for MediaConvert

To view examples of MediaConvert identity-based policies, see Identity-based policy examples for AWS Elemental MediaConvert.

Resource-based policies within MediaConvert

Supports resource-based policies: No

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. Adding a cross-account principal to a resource-based policy is only half of establishing the trust relationship. When the principal and the resource are in different AWS accounts, an IAM administrator in the trusted account must also grant the principal entity (user or role) permission to access the resource. They grant permission by attaching an identity-based policy to the entity. However, if a resource-based policy grants access to a principal in the same account, no additional identity-based policy is required. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

Policy actions for MediaConvert

Supports policy actions: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Policy actions usually have the same name as the associated AWS API operation.

There are some exceptions, such as *permission-only actions* that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

Include actions in a policy to grant permissions to perform the associated operation.

To see a list of MediaConvert actions, see Actions defined by AWS Elemental MediaConvert in the *Service Authorization Reference*.

Policy actions in MediaConvert use the following prefix before the action:

```
MediaConvert
```

To specify multiple actions in a single statement, separate them with commas.

```
"Action": [
      "MediaConvert:action1",
      "MediaConvert:action2"
         ]
```

To view examples of MediaConvert identity-based policies, see Identity-based policy examples for AWS Elemental MediaConvert.

Policy resources for MediaConvert

Supports policy resources: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. Statements must include either a `Resource` or a `NotResource` element. As a best practice, specify a resource using its Amazon Resource Name (ARN). You can do this for actions that support a specific resource type, known as *resource-level permissions*.

For actions that don't support resource-level permissions, such as listing operations, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

To see a list of MediaConvert resource types and their ARNs, see Resources defined by AWS Elemental MediaConvert in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see Actions defined by AWS Elemental MediaConvert.

To view examples of MediaConvert identity-based policies, see Identity-based policy examples for AWS Elemental MediaConvert.

Policy condition keys for MediaConvert

Supports service-specific policy condition keys: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Condition` element (or `Condition` *block*) lets you specify conditions in which a statement is in effect. The `Condition` element is optional. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request.

If you specify multiple `Condition` elements in a statement, or multiple keys in a single `Condition` element, AWS evaluates them using a logical `AND` operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical `OR` operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an IAM user permission to access a resource only if it is tagged with their IAM user name. For more information, see IAM policy elements: variables and tags in the *IAM User Guide*.

AWS supports global condition keys and service-specific condition keys. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

To see a list of MediaConvert condition keys, see Condition keys for AWS Elemental MediaConvert in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see Actions defined by AWS Elemental MediaConvert.

To view examples of MediaConvert identity-based policies, see Identity-based policy examples for AWS Elemental MediaConvert.

ACLs in MediaConvert

Supports ACLs: No

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

ABAC with MediaConvert

Supports ABAC (tags in policies): Partial

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes. In AWS, these attributes are called *tags*. You can attach tags to IAM entities (users or roles) and to many AWS resources. Tagging entities and resources is the first step of ABAC. Then you design ABAC policies to allow operations when the principal's tag matches the tag on the resource that they are trying to access.

ABAC is helpful in environments that are growing rapidly and helps with situations where policy management becomes cumbersome.

To control access based on tags, you provide tag information in the condition element of a policy using the `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

If a service supports all three condition keys for every resource type, then the value is **Yes** for the service. If a service supports all three condition keys for only some resource types, then the value is **Partial**.

For more information about ABAC, see What is ABAC? in the *IAM User Guide*. To view a tutorial with steps for setting up ABAC, see Use attribute-based access control (ABAC) in the *IAM User Guide*.

Using temporary credentials with MediaConvert

Supports temporary credentials: Yes

Some AWS services don't work when you sign in using temporary credentials. For additional information, including which AWS services work with temporary credentials, see AWS services that work with IAM in the *IAM User Guide*.

You are using temporary credentials if you sign in to the AWS Management Console using any method except a user name and password. For example, when you access AWS using your company's single sign-on (SSO) link, that process automatically creates temporary credentials. You also automatically create temporary credentials when you sign in to the console as a user and then switch roles. For more information about switching roles, see Switching to a role (console) in the *IAM User Guide*.

You can manually create temporary credentials using the AWS CLI or AWS API. You can then use those temporary credentials to access AWS. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM.

Forward access sessions for MediaConvert

Supports forward access sessions (FAS): Yes

When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.

Service roles for MediaConvert

Supports service roles: Yes

A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.

> **Warning**
> Changing the permissions for a service role might break MediaConvert functionality. Edit service roles only when MediaConvert provides guidance to do so.

Service-linked roles for MediaConvert

Supports service-linked roles: No

A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

For details about creating or managing service-linked roles, see [AWS services that work with IAM](). Find a service in the table that includes a `Yes` in the **Service-linked role** column. Choose the **Yes** link to view the service-linked role documentation for that service.

Identity-based policy examples for AWS Elemental MediaConvert

By default, users and roles don't have permission to create or modify MediaConvert resources. They also can't perform tasks by using the AWS Management Console, AWS Command Line Interface (AWS CLI), or AWS API. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

To learn how to create an IAM identity-based policy by using these example JSON policy documents, see Creating IAM policies in the *IAM User Guide*.

For details about actions and resource types defined by MediaConvert, including the format of the ARNs for each of the resource types, see Actions, resources, and condition keys for AWS Elemental MediaConvert in the *Service Authorization Reference*.

Topics

- Policy best practices
- Using the MediaConvert console
- Allow users to view their own permissions
- Example policy: Basic MediaConvert policy
- Example policy: Resource-level access control
- Example policy: Tag-based access control using resource tags

Policy best practices

Identity-based policies determine whether someone can create, access, or delete MediaConvert resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as AWS CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see IAM Access Analyzer policy validation in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Configuring MFA-protected API access in the *IAM User Guide*.

For more information about best practices in IAM, see Security best practices in IAM in the *IAM User Guide*.

Using the MediaConvert console

To access the AWS Elemental MediaConvert console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the MediaConvert resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won't function as intended for entities (users or roles) with that policy.

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that they're trying to perform.

To ensure that users and roles can still use the MediaConvert console, also attach the MediaConvert *`ConsoleAccess`* or *`ReadOnly`* AWS managed policy to the entities. For more information, see [Adding permissions to a user](#) in the *IAM User Guide*.

The following sample policy grants the IAM user permissions to all AWS Elemental MediaConvert actions (such as `ListJobs`, `CreateJob`, and so on) on all MediaConvert resources (such as jobs, queues, and output presets). It also grants the IAM actions required for the IAM user to specify the service role that MediaConvert will assume in order to run the job. It also grants Amazon S3 actions that allow the IAM user to use the **Browse** button to choose input and output locations. The Amazon S3 permissions aren't required to run the job; without them, the IAM user can specify the URL of the bucket instead. In this example, the account number is *`111122223333`* and the role name is *`MediaConvertRole`*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "mediaconvertActions",
      "Effect": "Allow",
      "Action": "mediaconvert:*",
      "Resource": "*"
    },
    {
      "Sid": "iamListRoles",
      "Effect": "Allow",
      "Action": "iam:ListRoles",
      "Resource": "*"
    },
    {
      "Sid": "iamPassRole",
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::111122223333:role/MediaConvertRole"
    },
```

```
    {
      "Sid": "s3Actions",
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:GetBucketLocation",
        "s3:ListAllMyBuckets"
      ],
      "Resource": "*"
    }
  ]
}
```

Allow users to view their own permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
```

```
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Example policy: Basic MediaConvert policy

The following example policy grants the basic permissions to operate AWS Elemental MediaConvert. In this example, the account number is *111122223333* and the role name is *MediaConvertRole*. If you are using encryption, or if your Amazon S3 buckets have default encryption enabled, you need additional permissions. For more information, see Protecting your media assets with encryption and DRM using AWS Elemental MediaConvert.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "mediaconvertActions",
      "Effect": "Allow",
      "Action": "mediaconvert:*",
      "Resource": "*"
    },
    {
      "Sid": "iamListRoles",
      "Effect": "Allow",
      "Action": "iam:ListRoles",
      "Resource": "*"
    },
    {
      "Sid": "iamPassRole",
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::111122223333:role/MediaConvertRole"
    },
    {
      "Sid": "s3Actions",
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
```

```
            "s3:GetBucketLocation",
            "s3:ListAllMyBuckets"
        ],
        "Resource": "*"
      }
    ]
}
```

Example policy: Resource-level access control

The following example policy grants permissions to certain AWS Elemental MediaConvert resources in your account. In this example, the account number is *012345678910*. It allows the following actions, in any partition and Region:

- View the details of all queues in the account at once.
- View all the jobs that have been submitted to the queue "myQueue" at once.
- Create a job and submit it to the queue "myQueue", referencing any presets with names that start with "allow" and referencing any job template.
- Create a job template referencing the queue "myQueue" and any presets with names that start with "allow".

Note

You can't grant cross-account permissions to AWS Elemental MediaConvert resources, such as queues, output presets, and jobs. You can grant cross-account permissions to your input and output media files stored in Amazon S3.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowListQueues",
            "Effect": "Allow",
            "Action": "mediaconvert:ListQueues",
            "Resource": "*"
        },
        {
```

```
            "Sid": "AllowListJobsInQueue",
            "Effect": "Allow",
            "Action": "mediaconvert:ListJobs",
            "Resource": "arn:*:mediaconvert:*:012345678910:queues/myQueue"
        },
        {
            "Sid": "AllowCreateLimitedJobs",
            "Effect": "Allow",
            "Action": [
                "mediaconvert:CreateJob",
                "mediaconvert:CreateJobTemplate"
            ],
            "Resource": [
                "arn:*:mediaconvert:*:012345678910:queues/myQueue",
                "arn:*:mediaconvert:*:012345678910:presets/allow*",
                "arn:*:mediaconvert:*:012345678910:jobTemplates/*"
            ]
        }
    ]
}
```

Example policy: Tag-based access control using resource tags

The following policy grants access to all actions on all MediaConvert resources in the account, unless the action listed under `DenyMediaConvertWithResourceTag` is tagged with the key `access` and has a value `denied` or a value that starts with the string "`deny`".

> **Note**
>
> This policy demonstrates the IAM principle that, in the case where policies both allow and deny action on a resource, the denial takes precedence. Therefore, the IAM user with this attached policy can't perform the denied actions, even if they have a different policy that allows administrative permissions to all MediaConvert resources.

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

```
            "Effect": "Allow",
            "Action": "mediaconvert:*",
            "Resource": "*"
        },
        {
            "Sid": "DenyMediaConvertWithResourceTag",
            "Effect": "Deny",
            "Action": [
                "mediaconvert:DeleteJobTemplate",
                "mediaconvert:GetQueue",
                "mediaconvert:UpdateQueue",
                "mediaconvert:DeleteQueue",
                "mediaconvert:GetPreset",
                "mediaconvert:ListTagsForResource",
                "mediaconvert:GetJobTemplate",
                "mediaconvert:UntagResource",
                "mediaconvert:UpdateJobTemplate",
                "mediaconvert:DeletePreset",
                "mediaconvert:TagResource",
                "mediaconvert:UpdatePreset"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aws:ResourceTag/access": [
                        "deny*",
                        "denied"
                    ]
                }
            }
        }
    ]
}
```

Cross-service confused deputy prevention

The *confused deputy* problem is a security issue. It occurs when an entity, that doesn't have permission to perform an action, can coerce a more-privileged entity to perform the action. In AWS, cross-service impersonation can result in the confused deputy problem. Cross-service impersonation can occur when one service (the *calling service*) calls another service (the *called* *service*). The calling service can be manipulated to use its permissions to act on another customer's resources in a way it should not otherwise have permission to access. To prevent this, AWS provides tools that help you protect your data for all services with service principals that have been given access to resources in your account.

We recommend using the [`aws:SourceArn`] and [`aws:SourceAccount`] global condition context keys in resource policies to limit the permissions that AWS Elemental MediaConvert gives another service to the resource. If the `aws:SourceArn` value does not contain the account ID you must use both global condition context keys to limit permissions. If you use both global condition context keys and the `aws:SourceArn` value contains the account ID, the `aws:SourceAccount` value and the account in the `aws:SourceArn` value must use the same account ID when used in the same policy statement. Use `aws:SourceArn` if you want only one resource to be associated with the cross-service access. Use `aws:SourceAccount` if you want to allow any resource in that account to be associated with the cross-service use.

The value of `aws:SourceArn` must be the ARN of the MediaConvert queue.

The most effective way to protect against the confused deputy problem is to use the `aws:SourceArn` global condition context key with the full ARN of the resource. If you don't know the full ARN of the resource or if you are specifying multiple resources, use the `aws:SourceArn` global context condition key with wildcards (*) for the unknown portions of the ARN. For example, `arn:aws:mediaconvert:*:123456789012:*`.

The following example shows how you can use the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in MediaConvert to prevent the confused deputy problem.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "mediaconvert.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:mediaconvert:*:123456789012:queues/Default"
      },
      "StringEquals": {
```

```
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

Troubleshooting AWS Elemental MediaConvert identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with MediaConvert and IAM.

Topics

- I am not authorized to perform an action in MediaConvert
- I am not authorized to perform iam:PassRole
- I want to allow people outside of my AWS account to access my MediaConvert resources

I am not authorized to perform an action in MediaConvert

If you receive an error that you're not authorized to perform an action, your policies must be updated to allow you to perform the action.

The following example error occurs when the `mateojackson` IAM user tries to use the console to view details about a fictional *`my-example-widget`* resource but doesn't have the fictional `MediaConvert:GetWidget` permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 MediaConvert:GetWidget on resource: my-example-widget
```

In this case, the policy for the `mateojackson` user must be updated to allow access to the *`my-example-widget`* resource by using the `MediaConvert:GetWidget` action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, your policies must be updated to allow you to pass a role to MediaConvert.

Some AWS services allow you to pass an existing role to that service instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named `marymajor` tries to use the console to perform an action in MediaConvert. However, the action requires the service to have permissions that are granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In this case, Mary's policies must be updated to allow her to perform the `iam:PassRole` action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I want to allow people outside of my AWS account to access my MediaConvert resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether MediaConvert supports these features, see How AWS Elemental MediaConvert works with IAM.
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see Providing access to AWS accounts owned by third parties in the *IAM User Guide*.
- To learn how to provide access through identity federation, see Providing access to externally authenticated users (identity federation) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.

Setting up access for other AWS accounts to your AWS Elemental MediaConvert outputs

When you follow the usual setup for permissions as described in Setting up IAM permissions , only users that belong to your AWS account can access your output files.

In some cases, you might want to allow users of other AWS accounts access to the outputs of your jobs. For example, you might run transcoding jobs on behalf of one of your customers, and you might want that customer to have access to the outputs of the jobs. You can do this in one of the following ways:

- **Grant access to your Amazon S3 bucket.**

  When you do this, MediaConvert writes your output files to your bucket, and you grant another account access to your bucket.
- **Have MediaConvert write your output files to a bucket owned by another account.**

  You can have MediaConvert apply an Amazon S3 canned access control list (ACL) to your outputs. A canned ACL is a predefined ACL that includes the necessary permissions.

  When you do this, you still own the output file, but MediaConvert grants access to the outputs to the owner of the bucket.

Topics

Granting access to your output Amazon S3 bucket

Suppose that you want the outputs of your MediaConvert jobs to reside in an Amazon S3 bucket that you own, but you want users that belong to another AWS account to have access to them. To grant access, you can add an Amazon S3 bucket policy to your output bucket.

For a tutorial about how to grant this access, see Example 2: Bucket owner granting cross-account bucket permissions in the *Amazon Simple Storage Service User Guide*.

The following example bucket policy grants access to your output bucket:

```
{
    "Id": "Policy1572454561447",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Stmt1572454547712",
            "Action": [
                "s3:GetObject"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
            "Principal": {
                "AWS": [
                    "111122223333"
                ]
            }
        }
    ]
}
```

Writing your outputs to an Amazon S3 bucket in another account

When you want the outputs of your MediaConvert jobs to reside in an Amazon S3 bucket that is owned by another AWS account, you work together with the administrator of that account to add a bucket permissions policy that grants you access to write the files and to add an Amazon S3 canned access control list (ACL) to the outputs. Then you set up your MediaConvert job to write to that bucket and to automatically add the canned ACL `bucket-owner-full-control` as it does so.

The result of this setup is that you own the files, but they reside in another account's bucket. The owner of the bucket has full access to the files.

To write your outputs to a bucket owned by another account

1. Work with an administrator of the other account to add a bucket policy to the Amazon S3 bucket that you want to write your output files to. For more information, see How do I add an S3 Bucket policy? in the *Amazon Simple Storage Service User Guide*.

   The following example bucket policy grants the necessary permissions:

```
{
    "Version": "2012-10-17",
    "Id": "Policy1570060985561",
    "Statement": [
        {
            "Sid": "Stmt1570060984261",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:role/MediaConvertRole"
                ]
            },
            "Action": [
                "s3:GetObject",
                "s3:GetObjectAcl",
                "s3:ListBucket",
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": [
                "arn:aws:s3:::bucket",
                "arn:aws:s3:::bucket/*"
            ]
        }
    ]
}
```

2. For any job that writes outputs to that bucket, apply the **Bucket owner full control** ACL as follows:

    a. Set up your job as usual. For more information, see Configuring jobs in MediaConvert.

    b. Specify the other account's Amazon S3 bucket for your output **Destination**. On the **Create job** page, in the **Job** pane on the left, choose an output group. In the group settings section on the right, find **Destination**.

    c. Enable **Access control**, and then choose **Bucket owner full control** for **Canned access control list**.

How to allow or disallow input location types

AWS Elemental MediaConvert supports Amazon S3, HTTPS, and HTTP input location types for your input media and files. You can allow or disallow access to one or more of these input location types by using a MediaConvert policy.

By default, each Region in your AWS account does not have a policy and MediaConvert allows all supported input location types. You only need to create an input policy if you want to disallow access to one or more of these input location types.

To prevent jobs from running with a disallowed input location type, create a MediaConvert *Input policy*.

Additionally, to prevent jobs from being submitted to the MediaConvert API if an Input policy isn't in place, create an IAM policy using *condition keys*. You can apply these IAM policies to IAM roles across your organization.

The following sections describe how to create an Input policy and how to use IAM condition keys to allow or disallow input location types.

Topics

How to allow or disallow input location types using an Input policy

To create or change a policy, submit a **put-policy** command using the API, SDK, or Command Line Interface (CLI) and include the policy in JSON. Visit the MediaConvert API Reference to learn more about supported policy commands and expected response codes.

The following is an example of how to submit a policy using the CLI. This example allows jobs with Amazon S3 and HTTPS inputs, and disallows jobs with HTTP inputs:

```
aws mediaconvert put-policy --policy '{"S3Inputs":"ALLOWED", "HttpsInputs":"ALLOWED",
 "HttpInputs":"DISALLOWED"}'
```

If you don’t specify an input location in the policy JSON, MediaConvert will treat the input location as ALLOWED. Here is another example that allows jobs with Amazon S3 and HTTPS inputs, and disallows jobs with HTTP inputs:

```
aws mediaconvert put-policy --policy '{"HttpInputs":"DISALLOWED"}'
```

Note that the put-policy command overwrites any existing policy in the Region.

Retrieve the current policy

To retrieve the current policy in JSON, submit a **get-policy** command:

```
aws mediaconvert get-policy
```

Delete the current policy

To delete the current policy and allow all inputs (reverting to the default behavior), submit a **delete-policy** command:

```
aws mediaconvert delete-policy
```

What happens when you try to submit a job with a disallowed input location?

If you attempt to submit a job that specifies an input location that your policy disallows, MediaConvert will instead return an HTTP 400 (BadRequestException) error. The error message will be: You specified an input location that your policy disallows. Specify an allowed input location and resubmit your job. Since MediaConvert prevents these jobs from being submitted, they will not appear in your job history.

If you submit a job that specifies an input location that is allowed, but the job requires accessing another input location that is disallowed, your job will fail. For example, you might encounter this if you specify an Apple HLS manifest on an allowed Amazon S3 location that references other input segment files on a disallowed HTTP location. The job failure error code will be 3457 and the message will be: You specified an input location that your policy disallows. Specify an allowed input location and resubmit your job.

How to use IAM condition keys with Input policies

When you include a *condition key* in your IAM policy that you use to submit create job requests, IAM checks if your account has an Input policy that matches that condition. The condition you specify must match your account's Input policy for the API request to be authorized. You can use any of the following boolean condition keys:

- **HttpInputsAllowed**
- **HttpsInputsAllowed**
- **S3InputsAllowed**

When using condition keys, consider the following scenarios:

If the condition and Input policy match, for example if you set **HTTPInputsAllowed** to *true* and your account's Input policy allows HTTP inputs, then your create job request will be submitted to the MediaConvert API.

If the condition and Input policy do not match, for example if you set **HTTPInputsAllowed** to *false* and your account's Input policy allows HTTP inputs, then your create job request will not be submitted to the MediaConvert API. You will receive following error message instead: "message": "User: arn:aws:iam::111122223333:user/User is not authorized to perform: mediaconvert:CreateJob on resource: arn:aws:mediaconvert:us-west-2:111122223333:queues/Default"

If the condition and Input policy match, for example if you set **HTTPInputsAllowed** to *false* and your account's Input policy *disallows* HTTP inputs, then your create job request will be submitted to the MediaConvert API. However, the API will then return an HTTP 400 (BadRequestException) error. The error message will be: You specified an input location that your policy disallows. Specify an allowed input location and resubmit your job.

For more information about using IAM condition keys, see IAM JSON policy elements: Condition in the *IAM User Guide*.

The following JSON is an example IAM policy using MediaConvert condition keys that checks if your account has an Input policy that disallows HTTP inputs:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "BlockHTTPInputsExample",
            "Effect": "Allow",
            "Action": "mediaconvert:CreateJob",
            "Resource": "*",
            "Condition": {
                "ForAllValues:BoolIfExists": {
```

```
                    "mediaconvert:HttpInputsAllowed": [
                        "false"
                    ],
                    "mediaconvert:HttpsInputsAllowed": [
                        "true"
                    ],

                    "mediaconvert:S3InputsAllowed": [
                        "true"
                    ]
                }
            }
        }
    ]
}
```

For more information about condition key support within MediaConvert, see How AWS Elemental MediaConvert works with IAM.

Compliance validation for AWS Elemental MediaConvert

Third-party auditors assess the security and compliance of AWS Elemental MediaConvert as part of multiple AWS compliance programs. These include SOC, PCI, FedRAMP, HIPAA, and others.

To learn whether an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program and choose the compliance program that you are interested in. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security and Compliance Quick Start Guides – These deployment guides discuss architectural considerations and provide steps for deploying baseline environments on AWS that are security and compliance focused.
- Architecting for HIPAA Security and Compliance on Amazon Web Services – This whitepaper describes how companies can use AWS to create HIPAA-eligible applications.

> **Note**
>
> Not all AWS services are HIPAA eligible. For more information, see the HIPAA Eligible Services Reference.

- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- AWS Customer Compliance Guides – Understand the shared responsibility model through the lens of compliance. The guides summarize the best practices for securing AWS services and map the guidance to security controls across multiple frameworks (including National Institute of Standards and Technology (NIST), Payment Card Industry Security Standards Council (PCI), and International Organization for Standardization (ISO)).
- Evaluating Resources with Rules in the *AWS Config Developer Guide* – The AWS Config service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS. Security Hub uses security controls to evaluate your AWS resources and to check your compliance against security industry standards and best practices. For a list of supported services and controls, see Security Hub controls reference.
- Amazon GuardDuty – This AWS service detects potential threats to your AWS accounts, workloads, containers, and data by monitoring your environment for suspicious and malicious activities. GuardDuty can help you address various compliance requirements, like PCI DSS, by meeting intrusion detection requirements mandated by certain compliance frameworks.
- AWS Audit Manager – This AWS service helps you continuously audit your AWS usage to simplify how you manage risk and compliance with regulations and industry standards.

Resilience in AWS Elemental MediaConvert

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between Availability Zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS global infrastructure.

Infrastructure security in AWS Elemental MediaConvert

As a managed service, AWS Elemental MediaConvert is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see AWS Cloud Security. To design your AWS environment using the best practices for infrastructure security, see Infrastructure Protection in *Security Pillar AWS Well‐Architected Framework*.

You use AWS published API calls to access MediaConvert through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

AWS Elemental MediaConvert related information

Find links to code examples, tutorials, and other helpful information for getting started with AWS Elemental MediaConvert.

Code examples and tutorials for AWS Elemental MediaConvert VOD solutions

Find code examples and tutorials for automating video on demand (VOD) workflows:

- **VOD automation toolkit** – example code and AWS CloudFormation scripts on GitHub for automating video on demand (VOD) workflows on AWS.
- **Video on demand at AWS** – example code, deployment guide, and AWS CloudFormation template in *AWS Solutions*. The workflow in this solution ingests source videos, processes the videos for playback on a wide range of devices, and stores the transcoded media files for on-demand delivery to end users through Amazon CloudFront.

Find code for a complete VOD workflow from ingest through delivery in Video on demand at AWS in *AWS Solutions*.

AWS learning resources

Explore the places to learn about using AWS services:

- Classes & Workshops – Links to role-based and specialty courses, in addition to self-paced labs to help sharpen your AWS skills and gain practical experience.
- AWS Developer Center – Explore tutorials, download tools, and learn about AWS developer events.
- AWS Developer Tools – Links to developer tools, SDKs, IDE toolkits, and command line tools for developing and managing AWS applications.
- Getting Started Resource Center – Learn how to set up your AWS account, join the AWS community, and launch your first application.
- Hands-On Tutorials – Follow step-by-step tutorials to launch your first application on AWS.
- AWS Whitepapers – Links to a comprehensive list of technical AWS whitepapers, covering topics such as architecture, security, and economics and authored by AWS Solutions Architects or other technical experts.

- AWS Support Center – The hub for creating and managing your AWS Support cases. Also includes links to other helpful resources, such as forums, technical FAQs, service health status, and AWS Trusted Advisor.
- AWS Support – The primary webpage for information about AWS Support, a one-on-one, fast-response support channel to help you build and run applications in the cloud.
- Contact Us – A central contact point for inquiries concerning AWS billing, account, events, abuse, and other issues.
- AWS Site Terms – Detailed information about our copyright and trademark; your account, license, and site access; and other topics.

Document history for User Guide

The following table describes important additions to the AWS Elemental MediaConvert documentation. We also update the documentation frequently to address the feedback that you send us.

Change	Description	Date
Search jobs	With AWS Elemental MediaConvert, you can search for jobs with a specific input file name.	June 20, 2024
Dolby Vision	MediaConvert added SDR and HDR10 input support for Dolby Vision Profile 8.1 outputs.	June 14, 2024
Audio description mixing	MediaConvert now supports audio description mixing.	March 6, 2024
Dolby Vision	MediaConvert added SDR input support for Dolby Vision outputs.	March 1, 2024
3D LUTs	MediaConvert added support for color conversion with 3D LUTs.	January 17, 2024
Video scaling behavior	MediaConvert added the following video scaling modes: **Fit**, **Fit without upscaling**, and **Fill**.	November 10, 2023
Accelerated transcoding	MediaConvert added accelerat ed transcoding support for Matroska inputs.	October 20, 2023

Video overlays	MediaConvert added support for video overlays.	October 4, 2023
Supported output formats	Documentation update. Combined video and audio tables. Added crosslinks to containers and codecs reference tables.	October 4, 2023
Supported inputs and outputs	Added containers and codecs reference tables.	October 4, 2023
Supported input formats	Documentation update. Combined video and audio tables. Added crosslinks to containers and codecs reference tables.	October 4, 2023
Documentation update	Reorganized the User Guide.	September 19, 2023
Queues	Reorganized queues documentation, including updated procedures and AWS CLI examples.	July 19, 2023
Color warnings	MediaConvert added missing color metadata and sample range warnings.	July 17, 2023
Dolby Vision	MediaConvert added Image inserter support for HDR10+ inputs and Dolby Vision outputs.	June 30, 2023

AAC audio codec	We added a section that lists the supported combinations of coding mode, profile, sampling rate, and bitrate for the AAC codec.	May 31, 2023
MXF profile XDCAM RDD9	MediaConvert added support for MXF outputs with XDCAM RDD9 profiles.	May 19, 2023
TLS	MediaConvert API clients now require a minimum of TLS 1.2.	May 12, 2023
Encoding settings for video quality	Added additional job settings recommendations for optimized video quality.	April 25, 2023
Video passthrough	MediaConvert now supports video passthrough for I-frame only input videos.	April 24, 2023
Troubleshooting	AWS Elemental MediaConvert now returns warnings when transcoding jobs run into problems that do not prevent the job from completing. See the new Troubleshooting chapter for information about both warnings and errors.	April 5, 2023
Monitoring AWS Elemental MediaConvert	AWS Elemental MediaConvert now emits more Amazon EventBridge events and Amazon CloudWatch metrics. Use them to help identify trends, statistics, or issues with your jobs and outputs.	April 5, 2023

Animated GIF input	MediaConvert added support for animated GIF as an input format.	February 28, 2023
AWS Identity and Access Management	Updated guide to align with the IAM best practices . For more information, see Security best practices in IAM.	February 14, 2023
VFR WEBM and MKV inputs	MediaConvert added input support for WEBM and MKV containers with variable frame rates.	February 1, 2023
Create Dolby Vision outputs from HDR10 inputs	MediaConvert added support for Dolby Vision Profile 5 outputs that are created from HDR10 inputs.	June 23, 2022
Automated ABR rules	MediaConvert added rendition size restriction rules for Automated ABR.	May 24, 2022
Video generator	MediaConvert added support for generating black video.	April 29, 2022
Pad video	MediaConvert added support for padding video with black video frames.	April 29, 2022
Create audio-only outputs in a DASH container	MediaConvert added support for audio-only DASH outputs	April 29, 2022
Create Dolby Vision profile 8.1 outputs	MediaConvert now supports Dolby Vision Profile 8.1 outputs, which are compatibl e with both HDR10 and Dolby Vision players.	April 29, 2022

Encoding settings for video quality	Learn how to create AWS Elemental MediaConvert jobs with encoding settings that are automatically optimized for video quality.	November 15, 2021
Allow or disallow input location types	You can now allow or disallow input location types using an Input policy.	October 8, 2021
TTML style formatting	Learn about style formatting for TTML input captions.	September 16, 2021
New partner integration for Kantar SNAP watermarking	Learn about using AWS Elemental MediaConvert to encode Kantar audio watermarks.	May 10, 2021
Add support for HLS ingest	MediaConvert now supports HLS as an input format.	March 31, 2021
Refined automatic MXF profile selection for MPEG-2 outputs	MediaConvert default automatic MXF profile selection now chooses the SMPTE-386 D10 profile when the output's encoding settings allow for it. Otherwise , MediaConvert chooses XDCAM RDD9. Learn the details about the required encoding settings for D10.	February 9, 2021
New supported input audio container: WMA	AWS Elemental MediaConv ert now supports WMA audio inputs.	February 9, 2021

Docs-only update: Remove HLS ingest support docs	This update corrects a documentation error that shows support for HLS ingest. MediaConvert will provide support for HLS ingest in an upcoming release.	February 9, 2021
AWS Elemental MediaConvert now supports HLS inputs	Learn about the package requirements, compatible features, and feature restrictions when you use an HLS package as input to MediaConvert.	February 9, 2021
HTTP inputs to AWS Elemental MediaConvert can now use redirects	See the updated requirements for providing your inputs to MediaConvert through HTTP.	January 7, 2021
AWS Elemental MediaConvert simplifies setting up permissions to use the service.	You can create an AWS Identity and Access Management (IAM) role to grant permissions to the service, by following these steps within the MediaConvert console.	January 7, 2021
Learn about using the new automated ABR feature.	With automated ABR, AWS Elemental MediaConvert sets up your adaptive bitrate (ABR) stack for you, based on your input. The service creates the right number of renditions, at the appropriate resolutions and bitrates.	November 11, 2020

Support for additional MXF profiles and automatic profile selection	Learn about creating MXF outputs with MediaConvert.	October 7, 2020
New partner integration for Nielsen non-linear watermark ing	Learn about using AWS Elemental MediaConvert to encode Nielsen audio watermarks.	October 7, 2020
Documentation-only update. New topics about frame rate conversion, telecine and inverse telecine, and scan type	Learn about the AWS Elemental MediaConvert settings for converting frame rate, adding and removing telecine, and converting scan type between progressive and interlaced.	October 7, 2020
New supported output container for VP8 and VP9 video	AWS Elemental MediaConv ert now supports the VP8 and VP9 video codec in DASH package outputs, in addition to the previously existing support in WebM.	August 27, 2020
New supported input container for Opus audio	AWS Elemental MediaConv ert now supports Opus audio inputs in the Matroska container, both as sidecar audio with a video output and as audio-only input. This is in addition to existing support of Opus inputs in the WebM container.	August 27, 2020

New supported output formats	Updated output codec and container reference table to reflect new format support. MediaConvert now supports WebM as an output container with VP8 and VP9 video codecs and Opus and Vorbis audio codecs.	June 3, 2020
Queue hopping	Learn about the AWS Elemental MediaConvert queue hopping feature. This feature lets you set up a job to automatically move to a different queue after waiting a specified length of time in the queue that you originally submitted it to.	April 8, 2020
You can now convert from HDR to SDR with automatic tone mapping	Learn about how AWS Elemental MediaConvert uses tone mapping to automatically convert from HDR formats to SDR color spaces.	March 6, 2020
8k output support in AWS Elemental MediaConvert	MediaConvert now supports up to 8k output resolution. Find more information about resolution maximums by output codec.	November 25, 2019
Use accelerated transcoding with slow PAL	You can now use accelerated transcoding for slow PAL jobs.	November 15, 2019

Set up access for other AWS accounts to your MediaConvert outputs	You can now grant cross-account access by writing your AWS Elemental MediaConvert outputs to an Amazon S3 bucket owned by another AWS account and applying a canned access control list (ACL) to your outputs. This chapter also provides information about the alternative method for setting up access—granting other accounts access to your output bucket in Amazon S3.	November 15, 2019
New supported input format	Updated reference tables to reflect new input format support. MediaConvert now supports HEVC (H.265) in QuickTime containers as input files.	November 15, 2019
New supported input captions format: SMPTE-TT	Updated reference tables reflect new captions support. MediaConvert now supports SMPTE-TT captions as input.	November 15, 2019

New supported captions format	Updated reference tables to reflect new captions support. IMSC sidecar output captions now available in **CMAF**, **DASH ISO**, and **File group** output groups. You can generate these from the following input captions formats: Ancillary, Embedded, IMSC, Teletext, TTML, SCC, SCTE-20, SMI, SRT, and STL.	November 15, 2019
ESAM support in DASH outputs	You can now specify ad insertion points in DASH outputs by including Event Signaling and Managemen t (ESAM) XML documents as strings inside your AWS Elemental MediaConvert job settings.	November 15, 2019
Doc-only update: Synchroni zing sidecar captions and audio in jobs that use input clipping and stitching	Learn how MediaConvert input and output timelines synchronize video, audio, captions, and overlays in assembly workflow jobs.	November 15, 2019
Creating additional manifests	You can now set up your ABR streaming packages in MediaConvert with addi tional top-level manifests that specify different subsets of your outputs.	November 15, 2019
Create Dolby Vision outputs	Learn how to create Dolby Vision outputs with MediaConvert.	November 15, 2019

Use integrated AWS tags on AWS Elemental MediaConvert resources	You can now use standard AWS tags with MediaConvert jobs the same way that you do with queues, templates , and presets. You can use these tags to sort your billing report for cost allocation purposes, to include your MediaConvert resources in resource groups with other AWS resources, and to control access to specific resources.	October 9, 2019
Use accelerated transcoding with frame capture	You can now use accelerat ed transcoding for jobs that include frame capture outputs.	October 9, 2019
Have AWS Elemental MediaConvert automatic ally choose accelerated transcoding for jobs that are compatible	You can now set accelerated transcoding to PREFERRED , to have the service fall back to standard transcodi ng when your inputs or job settings aren't compatible wi th accelerated transcoding.	October 9, 2019
Doc-only update, use cases for **Time delta**	Learn about using the **Time delta** setting to synchronize sidecar captions with your video. See use case examples that illustrate common captions sync problems that you can fix with this setting.	September 6, 2019

Choose the encryption method for DRM in CMAF output groups	You can now choose between AES-CTR and AES-CBC subsample encryption types when you enable DRM for CMAF output groups. Learn how to set up DRM encryption.	September 6, 2019
Align sidecar captions with your video after using the Timecode source setting	When you use the input setting **Timecode source** to align your sidecar audio file with your video, make sure that your sidecar captions still line up. Learn which sidecar captions formats require adjustment and how to make any necessary adjustments.	September 6, 2019
Multi-system SPEKE DRM for CMAF	When you work with SPEKE-compliant DRM partners to enable DRM on your CMAF outputs, you can now signal up to three system IDs in your DASH manifest and one in your HLS manifest. Learn about setting up this SPEKE encryption parameter in your AWS Elemental MediaConvert job.	August 12, 2019
Export and import jobs	Learn how to export MediaConvert jobs and how to create new jobs by importing one.	August 9, 2019

Specify the priority of your jobs	Learn how to set the relative priority of your jobs when you create them.	July 30, 2019
Convert two SCC input files to embedded captions	Learn about converting dual SCC input captions to embedded format.	July 30, 2019
Create audio-only outputs	Learn how to create MediaConvert outputs that contain only audio, without video.	July 23, 2019
HTTP inputs to AWS Elemental MediaConvert	You can now specify an HTTP(S) URL for your primary inputs to MediaConvert transcoding jobs. Learn about HTTP(S) input requirements.	July 12, 2019
Monitor output file upload progress	Using CloudWatch Events, you can monitor the progress of your job with finer granularity. That's because now AWS Elemental MediaConvert reports percentage completion of the job phases PROBING, UPLOADING, and TRANSCODING. This is in addition to reporting overall job progress in percent completion.	May 7, 2019

Doc-only update, format identifiers	Learn how to use format identifiers as variables in your job settings. For example, you can use a format identifie r to represent your input file name in a job template or output preset, so that AWS Elemental MediaConvert uses the input file name in the file path for your output.	April 24, 2019
You can now cancel an in-progress job	With AWS Elemental MediaConvert, you can cancel jobs that are already in progress.	March 13, 2019
Monitor job progress	With CloudWatch Events, you can now monitor AWS Elemental MediaConvert job progress by viewing the percentage of job completion. Monitoring your job's progress helps you maintain its reliabili ty, availability, and performan ce.	March 13, 2019
Find information about the new rotate feature.	For most inputs, you can now use the rotate feature to specify how the service rotates your video from input to output. You can also specify whether the service follows any rotation metadata in the input. Find information and instructions for setting rotation.	February 19, 2019

ESAM support	You can now specify ad insertion points by including Event Signaling and Management (ESAM) XML documents as strings inside your AWS Elemental MediaConvert job settings.	February 13, 2019
Code examples, example workflows, and other related information	Find links to code examples, tutorials, and other helpful information for getting started with AWS Elemental MediaConvert.	February 13, 2019
New supported captions formats	Updated reference tables to reflect new support for the following captions formats. On input: SMI, SCTE-20. On output: SCTE-20 + embedded, Embedded + SCTE-20, SMI.	November 19, 2018
Find information about the new image inserter features	You can now use the image inserter for still image overlays on individual inputs and outputs and for motion image overlays in AWS Elemental MediaConvert. This guide shows you how.	November 19, 2018

Added content key encryption to DRM encryption	Added the option to encrypt content keys. Prior to this, AWS Elemental MediaConv ert supported plaintext key delivery only. To use content key encryption, your DRM key provider must support encrypted content keys. If you enable this feature for a key provider that doesn't handle content key encryption, the operation fails.	November 19, 2018
Reserved queues	Learn about the new pricing option, reserved queues, in AWS Elemental MediaConv ert. With reserved queues, you pay for the transcoding capacity of the entire queue, regardless of how much or how little you use it. If you continuously or frequentl y run transcoding jobs, a reserved queue can help reduce costs.	September 27, 2018
Find output file names and paths in EventBridge notificat ions	You can receive output file names and paths, including manifest and media file outputs, by using Amazon EventBridge with AWS Elemental MediaConvert jobs.	September 18, 2018

Set up cost allocation reports to sort your AWS bill by the tags that you apply to AWS Elemental MediaConvert resources	You can use the tags you apply to AWS Elemental MediaConvert queues, job templates, and output presets as cost allocation tags. First activate them in the AWS Billing and Cost Managemen t dashboard, and then set up a billing report to view your AWS charges by resource.	September 7, 2018
Quality-defined variable bitrate (QVBR) mode now available	You can now use the QVBR rate control mode to get better video quality for the same file size, or to reduce your file sizes while maintaining video quality. Get information about how to set it up.	August 13, 2018
Use AWS Elemental MediaConvert tags for cost allocation through tagging	You can now activate tags on AWS Elemental MediaConvert queues, job templates, and output presets in the AWS Billing and Cost Managemen t dashboard, and then set up a monthly cost allocation report.	July 31, 2018
Tagging AWS Elemental MediaConvert resources on the console	You can now work with tags on existing AWS Elemental MediaConvert queues, job templates, and output presets in the MediaConvert console.	July 31, 2018

Tagging AWS Elemental MediaConvert resources initial release	You can now tag MediaConvert resources when you create them using the MediaConvert console, the MediaConvert API, or the AWS CLI. You can list the tags on an existing MediaConvert resource, and add and remove tags on existing MediaConvert resources, through the MediaConvert API and AWS CLI.	July 16, 2018
CMAF support	MediaConvert adds support for common media application format (CMAF) outputs.	May 4, 2018
New MediaConvert service release	Initial documentation for the MediaConvert service.	November 27, 2017

Note

- The AWS Media Services are not designed or intended for use with applications or in situations requiring fail-safe performance, such as life safety operations, navigation or communication systems, air traffic control, or life support machines in which the unavailability, interruption or failure of the services could lead to death, personal injury, property damage or environmental damage.
- A component of MediaConvert 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 MediaConvert 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.

- MediaConvert may contain Dolby Digital and Dolby Digital Plus, which are protected under international and U.S. copyright laws as unpublished works. Dolby Digital and Dolby Digital Plus are confidential and proprietary to Dolby Laboratories. Their reproduction or disclosure, in whole or in part, or the production of derivative works therefrom without the express permission of Dolby Laboratories is prohibited. © Copyright 2003-2015 Dolby Laboratories. All rights reserved.

AWS Glossary

For the latest AWS terminology, see the AWS glossary in the *AWS Glossary Reference*.