## Table of Contents

Getting Started with the SDKs and CLI ................................................................. 1
Getting Started with the API .................................................................................. 27
Postman Collection Files ...................................................................................... 28
GET Collection ...................................................................................................... 28
POST Collection .................................................................................................... 30
Sample Settings in JSON ...................................................................................... 34
Simple Example ...................................................................................................... 34
Complex Example .................................................................................................. 36
Important Notes ..................................................................................................... 43
Versioned Endpoints ............................................................................................ 43
Using the AWS CLI ................................................................................................ 43
About the Resources Section of This Guide ....................................................... 43
Resources ................................................................................................................ 44
Endpoints ................................................................................................................ 44
  URI ...................................................................................................................... 44
  HTTP Methods .................................................................................................. 44
  Schemas ............................................................................................................. 45
  Properties .......................................................................................................... 45
JobTemplates ......................................................................................................... 46
  URI ...................................................................................................................... 46
  HTTP Methods .................................................................................................. 46
  Schemas ............................................................................................................. 47
  Properties .......................................................................................................... 79
JobTemplates name .............................................................................................. 198
  URI ...................................................................................................................... 198
  HTTP Methods .................................................................................................. 199
  Schemas ............................................................................................................. 200
  Properties .......................................................................................................... 232
Jobs ......................................................................................................................... 351
  URI ...................................................................................................................... 351
  HTTP Methods .................................................................................................. 351
  Schemas ............................................................................................................. 352
  Properties .......................................................................................................... 384
Jobs id ..................................................................................................................... 507
  URI ...................................................................................................................... 507
  HTTP Methods .................................................................................................. 507
  Schemas ............................................................................................................. 508
  Properties .......................................................................................................... 519
Presets ..................................................................................................................... 640
  URI ...................................................................................................................... 640
  HTTP Methods .................................................................................................. 640
  Schemas ............................................................................................................. 641
  Properties .......................................................................................................... 663
Presets name ......................................................................................................... 754
  URI ...................................................................................................................... 754
  HTTP Methods .................................................................................................. 754
  Schemas ............................................................................................................. 756
  Properties .......................................................................................................... 778
Queues .................................................................................................................... 868
  URI ...................................................................................................................... 868
  HTTP Methods .................................................................................................. 868
  Schemas ............................................................................................................. 869
  Properties .......................................................................................................... 870
Queues name ....................................................................................................... 874
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI</td>
<td>874</td>
</tr>
<tr>
<td>HTTP Methods</td>
<td>874</td>
</tr>
<tr>
<td>Schemas</td>
<td>875</td>
</tr>
<tr>
<td>Properties</td>
<td>877</td>
</tr>
<tr>
<td>Document History</td>
<td>880</td>
</tr>
<tr>
<td>AWS Glossary</td>
<td>881</td>
</tr>
</tbody>
</table>
Getting Started with AWS Elemental MediaConvert Using the AWS SDKs or the AWS CLI

To get started with AWS Elemental MediaConvert using one of the AWS SDKs or the AWS Command Line Interface (AWS CLI), follow this general procedure. For specific instructions and examples, choose a language in the final step of this procedure.

1. Set up AWS Identity and Access Management (IAM) permissions for both yourself and for the AWS Elemental MediaConvert service to access your resources on your behalf:
   - For information about setting up permissions for yourself, see the IAM User Guide.
   - For information about setting up permissions for the service to access your resources, see the Set Up IAM Permissions topic of the AWS Elemental MediaConvert User Guide.

2. In your client configuration, specify your authentication credentials and your AWS Region. For instructions that are specific to the programming language that you use, choose from this list of links to open the relevant topics in the AWS CLI or SDK guides:
   - AWS CLI
   - C++: credentials and region
   - Java
   - JavaScript
   - .NET
   - PHP
   - Python: credentials and region
   - Ruby
   - Tools for PowerShell

3. Get your account-specific endpoint and send your AWS Elemental MediaConvert requests to it. With most AWS services, you send your service request to a public endpoint. But with AWS Elemental MediaConvert, you request an endpoint that is specific to your account, and then you send your service requests to that.

   For specific instructions and code samples, choose one of the following programming languages:

   **AWS CLI**

   To create a transcoding job using the AWS Command Line Interface:

   1. Use `describe-endpoints` to get your account endpoint and set your region. In this example, the region is set to `ap-northeast-3`:

      ```
      aws mediaconvert describe-endpoints --region ap-northeast-3
      ```

   2. Use the `--endpoint-url` option to send your request to your account endpoint:

      ```
      aws --endpoint-url=https://abcd1234.mediaconvert.region-name-1.amazonaws.com --region=region-name-1 mediaconvert create-job --cli-input-json=file://~/job.json
      ```
In the preceding example, job.json specifies your transcoding job settings. You can use the AWS Elemental MediaConvert console to generate the JSON job specification by choosing your job settings, and then choosing Show job JSON at the bottom of the Job section. For sample job specifications, see Sample Settings in JSON (p. 34).

C++

To send requests using the SDK for C++:

```c++
*/

#include "stdafx.h" // do not include this line in the example :)

#include <aws/core/Aws.h>
#include <aws/core/utils/Outcome.h>
#include <aws/mediaconvert/MediaConvertClient.h>
#include <aws/mediaconvert/Model/DescribeEndpointsRequest.h>
#include <aws/mediaconvert/Model/CreateJobRequest.h>
#include <aws/mediaconvert/Model/CreateJobResult.h>

/* ----------------------------------------------
* Permissions IAM user needs to run this example
* ----------------------------------------------
*
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": ["mediaconvert:DescribeEndpoints", "mediaconvert:CreateJob"],
            "Resource": "*"
        }
    ]
}

/*
* JSON job settings used in this example
* --------------------------------------
*
{
    "UserMetadata": {
        "Customer": "Amazon"
    },
    "Role": "Your AWS Elemental MediaConvert role ARN",
    "Settings": {
        "OutputGroups": [
            {
                "Name": "File Group",
                "OutputGroupSettings": {
                    "Type": "FILE_GROUP_SETTINGS",
                    "FileGroupSettings": {
                        "Destination": "s3://youroutputdestination"
                    }
                }
            }
        ]
    }
}
```
"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": 90,
                    "Slices": 1,
                    "GopBReference": "DISABLED",
                    "SlowPal": "DISABLED",
                    "SpatialAdaptiveQuantization": "DISABLED",
                    "TemporalAdaptiveQuantization": "DISABLED",
                    "FlickerAdaptiveQuantization": "DISABLED",
                    "EntropyEncoding": "CABAC",
                    "Bitrate": 5000000,
                    "FramerateControl": "SPECIFIED",
                    "RateControlMode": "CBR",
                    "CodecProfile": "MAIN",
                    "Telecine": "NONE",
                    "MinIInterval": 0,
                    "AdaptiveQuantization": "HIGH",
                    "CodecLevel": "AUTO",
                    "FieldEncoding": "PALL",
                    "QualityTuningLevel": "SINGLE_PASS",
                    "FramerateConversionAlgorithm": "DUPLICATE_DROP",
                    "UnregisteredSeiTimecode": "DISABLED",
                    "GopSizeUnits": "FRAMES",
                    "ParControl": "SPECIFIED",
                    "NumberBFramesBetweenReferenceFrames": 2,
                    "RepeatPps": "DISABLED",
                    "FramerateDenominator": 30,
                    "ParNumerator": 1,
                    "ParDenominator": 1
                }
            },
            "AfdSignaling": "NONE",
            "DropFrameTimecode": "ENABLED",
            "RespondToAfd": "NONE",
            "ColorMetadata": "INSERT"
        },
        "AudioDescriptions": [
            {
                "AudioTypeControl": "FOLLOW_INPUT",
                "CodecSettings": {
                    "Codec": "AAC",
                    "AacSettings": {
                        "AudioDescriptionBroadcasterMix": "NORMAL",
                        "RateControlMode": "CBR",
                        "CodecProfile": "LC",
                        "CodingMode": "CODING_MODE_2_0",
                        "RawFormat": "NONE",
                        "SampleRate": 48000,
                        "Specification": "MPEG4",
                        "Bitrate": 64000
                    }
                }
            }
        ]
    },
int main()
{
    // Initialize the C++ SDK
    Aws::SDKOptions options;
    Aws::InitAPI(options);

    std::string mediaConvertRole = "arn:aws:iam::640773029566:role/media-convert-role";
    std::string fileInput = "s3://media-convert-sample/my-video.mp4";
    std::string fileOutput = "s3://media-convert-sample";
    // Once you know what your customer endpoint is, set it here
    std::string mediaConvertEndpoint = "";
    // If we do not have our customer-specific endpoint
if (mediaConvertEndpoint.empty())
{
// Obtain the customer-specific MediaConvert endpoint
Aws::Client::ClientConfiguration clientConfig;
clientConfig.region = "us-west-2";
Aws::MediaConvert::MediaConvertClient client(clientConfig);
Aws::MediaConvert::Model::DescribeEndpointsRequest request;
// need to strip https:// from endpoint for C++
mediaConvertEndpoint =
client.DescribeEndpoints(request).GetResult().GetEndpoints().at(0).GetUrl().substr(8);
}

// Create MediaConvert Client with the endpoints and region from above
Aws::Client::ClientConfiguration mcClientConfig;
// also need to set region endpoint, must match endpoint embedded in custom endpoint
mcClientConfig.region = "us-west-2";
mcClientConfig.endpointOverride = mediaConvertEndpoint;
Aws::MediaConvert::MediaConvertClient mcClient(mcClientConfig);

// Create Job request
Aws::MediaConvert::Model::CreateJobRequest createJobRequest;
createJobRequest.SetRole(mediaConvertRole);
Aws::Http::HeaderValueCollection hvc;
hvc.emplace("Customer", "Amazon");
createJobRequest.SetUserMetadata(hvc);

// Create job settings
Aws::MediaConvert::Model::JobSettings jobSettings;
jobSettings.SetAdAvailOffset(0);
Aws::MediaConvert::Model::TimecodeConfig timecodeConfig;
timecodeConfig.SetSource(Aws::MediaConvert::Model::TimecodeSource::EMBEDDED);
jobSettings.SetTimecodeConfig(timecodeConfig);
createJobRequest.SetSettings(jobSettings);

// Output Group
Aws::MediaConvert::Model::OutputGroup og;
og.SetName("File Group");
Aws::MediaConvert::Model::OutputGroupSettings ogs;
ogs.SetType(Aws::MediaConvert::Model::OutputGroupType::FILE_GROUP_SETTINGS);
Aws::MediaConvert::Model::FileGroupSettings fgs;
fgs.SetDestination(fileOutput);
ogs.SetFileGroupSettings(fgs);
og.SetOutputGroupSettings(ogs);

Aws::MediaConvert::Model::Output output;
output.SetNameModifier("_1");

Aws::MediaConvert::Model::VideoDescription vdes;
vdes.SetScalingBehavior(Aws::MediaConvert::Model::ScalingBehavior::DEFAULT);
vdes.SetTimecodeInsertion(Aws::MediaConvert::Model::VideoTimecodeInsertion::DISABLED);
vdes.SetAntiAlias(Aws::MediaConvert::Model::AntiAlias::ENABLED);
vdes.SetSharpness(50);
vdes.SetAfdSignaling(Aws::MediaConvert::Model::AfdSignaling::NONE);
vdes.SetDropFrameTimecode(Aws::MediaConvert::Model::DropFrameTimecode::ENABLED);
vdes.SetRespondToAfd(Aws::MediaConvert::Model::RespondToAfd::NONE);
vdes.SetColorMetadata(Aws::MediaConvert::Model::ColorMetadata::INSERT);

Aws::MediaConvert::Model::VideoCodecSettings vcs;
vcs.SetCodec(Aws::MediaConvert::Model::VideoCodec::H_264);
vcs.SetNumberReferenceFrames(3);
vcs.SetSyntax(Aws::MediaConvert::Model::H264Syntax::DEFAULT);
h264.SetSoftness(0);
h264.SetGopClosedCadence(1);
h264.SetGopSize(90);
h264.SetSlices(1);
h264.SetGopBReference(Aws::MediaConvert::Model::H264GopBReference::DISABLED);
h264.SetSlowPal(Aws::MediaConvert::Model::H264SlowPal::DISABLED);
h264.SetSpatialAdaptiveQuantization(Aws::MediaConvert::Model::H264SpatialAdaptiveQuantization::ENABLED);
h264.SetTemporalAdaptiveQuantization(Aws::MediaConvert::Model::H264TemporalAdaptiveQuantization::ENABLED);
h264.SetFlickerAdaptiveQuantization(Aws::MediaConvert::Model::H264FlickerAdaptiveQuantization::DISABLED);
h264.SetEntropyEncoding(Aws::MediaConvert::Model::H264EntropyEncoding::CABAC);
h264.SetBitrate(5000000);
h264.SetFramerateControl(Aws::MediaConvert::Model::H264FramerateControl::SPECIFIED);
h264.SetRateControlMode(Aws::MediaConvert::Model::H264RateControlMode::CBR);
h264.SetCodecProfile(Aws::MediaConvert::Model::H264CodecProfile::MAIN);
h264.SetTelecine(Aws::MediaConvert::Model::H264Telecine::NONE);
h264.SetMinIInterval(0);
h264.SetAdaptiveQuantization(Aws::MediaConvert::Model::H264AdaptiveQuantization::HIGH);
h264.SetCodecLevel(Aws::MediaConvert::Model::H264CodecLevel::AUTO);
h264.SetFramerateConversionAlgorithm(Aws::MediaConvert::Model::H264FramerateConversionAlgorithm::DUPLICATE_DROP);
h264.SetUnregisteredSeiTimecode(Aws::MediaConvert::Model::H264UnregisteredSeiTimecode::DISABLED);
h264.SetGopSizeUnits(Aws::MediaConvert::Model::H264GopSizeUnits::FRAMES);
h264.SetParControl(Aws::MediaConvert::Model::H264ParControl::SPECIFIED);
h264.SetNumberBFramesBetweenReferenceFrames(2);
h264.SetRepeatPps(Aws::MediaConvert::Model::H264RepeatPps::DISABLED);
h264.SetFramerateNumerator(30);
h264.SetFramerateDenominator(1);
h264.SetParNumerator(1);
h264.SetParDenominator(1);
output.SetVideoDescription(vdes);

Aws::MediaConvert::Model::AudioDescription ades;

// This name matches one specified in the Inputs below
ades.SetLanguageCodeControl(Aws::MediaConvert::Model::AudioLanguageCodeControl::FOLLOW_INPUT);
ades.SetAudioSourceName("Audio Selector 1");

Aws::MediaConvert::Model::AudioCodecSettings acs;
acs.SetCodec(Aws::MediaConvert::Model::AudioCodec::AAC);

// This name matches one specified in the Inputs below
acs.SetAudioSourceName("Audio Selector 1");

Aws::MediaConvert::Model::AacSettings aac;
aac.SetAudioDescriptionBroadcasterMix(Aws::MediaConvert::Model::AacAudioDescriptionBroadcasterMix::NORMAL);
aac.SetRateControlMode(Aws::MediaConvert::Model::AacRateControlMode::CBR);
aac.SetCodecProfile(Aws::MediaConvert::Model::AacCodecProfile::LC);
aac.SetCodingMode(Aws::MediaConvert::Model::AacCodingMode::CODING_MODE_2_0);
aac.SetRawFormat(Aws::MediaConvert::Model::AacRawFormat::NONE);
aac.SetSampleRate(48000);
aac.SetSpecification(Aws::MediaConvert::Model::AacSpecification::MPEG4);
aac.SetBitrate(64000);
acs.SetAacSettings(aac);
ades.SetCodecSettings(acs);
output.SetAudioDescriptions(adess);
Aws::MediaConvert::Model::ContainerSettings mp4container;
mp4container.SetContainer(Aws::MediaConvert::Model::ContainerType::MP4);
Aws::MediaConvert::Model::Mp4Settings mp4;
mp4.SetCslgAtom(Aws::MediaConvert::Model::Mp4CslgAtom::INCLUDE);
mp4.SetFreeSpaceBox(Aws::MediaConvert::Model::Mp4FreeSpaceBox::EXCLUDE);
mp4.SetMoovPlacement(Aws::MediaConvert::Model::Mp4MoovPlacement::PROGRESSIVE_DOWNLOAD);
mp4container.SetMp4Settings(mp4);
og.AddOutputs(output);
jobSettings.AddOutputGroups(og);

// End Output Group

Aws::MediaConvert::Model::Input input;
input.SetFilterEnable(Aws::MediaConvert::Model::InputFilterEnable::AUTO);
input.SetPsiControl(Aws::MediaConvert::Model::InputPsiControl::USE_PSI);
input.SetFilterStrength(0);
input.SetDeblockFilter(Aws::MediaConvert::Model::InputDeblockFilter::DISABLED);
input.SetDenoiseFilter(Aws::MediaConvert::Model::InputDenoiseFilter::DISABLED);
input.SetTimecodeSource(Aws::MediaConvert::Model::InputTimecodeSource::EMBEDDED);
input.SetFileInput(fileInput);

Aws::MediaConvert::Model::AudioSelector audsel;
audsel.SetOffset(0);
audsel.SetDefaultSelection(Aws::MediaConvert::Model::AudioDefaultSelection::NOT_DEFAULT);
audsel.SetProgramSelection(1);
audsel.SetSelectorType(Aws::MediaConvert::Model::AudioSelectorType::TRACK);
audsel.AddTracks(1);
input.AddAudioSelectors("Audio Select 1", audsel);

Aws::MediaConvert::Model::VideoSelector vidsel;
vidsel.SetColorSpace(Aws::MediaConvert::Model::ColorSpace::FOLLOW);
input.SetVideoSelector(vidsel);

jobSettings.AddInputs(input);
createJobRequest.SetSettings(jobSettings);

Aws::MediaConvert::Model::CreateJobOutcome createJobResponse = mcClient.CreateJob(createJobRequest);
std::cout << createJobResponse.GetResult().GetJob().GetId() << std::endl;

Aws::ShutdownAPI(options);
return 0;

Go

After you use the DescribeEndpoints method to request an account-specific endpoint, send your requests to it as described in the Creating an Account Endpoint topic in the AWS SDK for Go Developer Guide.

Important
Make the DescribeEndpoints call only once in your application. Don't use DescribeEndpoints to create your AWS client each time that you make a request to
AWS Elemental MediaConvert. Otherwise, you will reach the throttle maximum on the public API endpoint.

Java

For information about setting up your credentials and region in your client configuration, see the Set up AWS Credentials and Region for Development topic in the AWS SDK for Java Developer Guide.

This procedure shows you how to get your account-specific endpoint and send AWS Elemental MediaConvert requests to it.

1. Add the following import statements:

```java
import com.amazonaws.client.builder.AwsClientBuilder.EndpointConfiguration;
import com.amazonaws.regions.Region;
import com.amazonaws.services.mediaconvert.*;
```

2. Determine the region for your endpoint and create an AWSMediaConvert client object for it:

```java
String region = "us-west-2";
AWSMediaConvert mediaConvertClient =
AWSMediaConvertClientBuilder.standard()
    .withRegion(region)
    .build();
```

3. Call the describeEndpoints method to retrieve the endpoint and save the endpoint's URL:

```java
DescribeEndpointsRequest request = new DescribeEndpointsRequest();
String endpoint = mediaConvertClient.describeEndpoints(request)
    .getEndpoints()
    .get(0).getUrl();
```

Important
Make the DescribeEndpoints call only once in your application. Don't use DescribeEndpoints to create your AWS client each time that you make a request to AWS Elemental MediaConvert. Otherwise, you will reach the throttle maximum on the public API endpoint.

4. Create a job request and a submit job request object:

```java
CreateJobRequest jobParam = new CreateJobRequest()
    .withSettings(jobSettings);
CreateJobResult mcResponse = new CreateJobResult();
mcResponse = mcClient.createJob(jobParam)
```

Note
The jobSettings object contains settings parameters. For sample JSON file job specifications, see Sample Settings in JSON (p. 34).

JavaScript

To send requests using the AWS SDK for JavaScript:

1. Get your account-specific endpoint.
Create a Node.js module with the file name `emc_getendpoint.js`.

Create an object to pass the empty request parameters for the `describeEndpoints` method of the `AWS.MediaConvert` client class. To call the `describeEndpoints` method, create a promise for invoking an AWS Elemental MediaConvert service object, passing the parameters. Then handle the response in the promise callback.

**Important**

Make the `DescribeEndpoints` call only once in your application. Don’t use `DescribeEndpoints` to create your AWS client each time that you make a request to AWS Elemental MediaConvert. Otherwise, you will reach the throttle maximum on the public API endpoint.

```javascript
// Load the AWS SDK for Node.js
var AWS = require('aws-sdk');

// Set the region
AWS.config.update({ region: 'us-west-2' });

// Create empty request parameters
var params = {
  MaxResults: 0,
};

// Create a promise on a MediaConvert object
var endpointPromise = new AWS.MediaConvert({ apiVersion: '2017-08-29' }).describeEndpoints(params).promise();

endpointPromise.then(
  function (data) {
    console.log("Your MediaConvert endpoint is ", data.Endpoints);
  },
  function (err) {
    console.log("Error", err);
  }
);
```

To run the example, type the following at the command line:

```bash
node ec2_getendpoint.js
```

2. Configure the SDK.

Configure the SDK for JavaScript by creating a global configuration object, and then setting the region for your code. In this example, the region is set to `us-west-2`. Because AWS Elemental MediaConvert uses endpoints that are unique to each AWS account, you must also configure the `AWS.MediaConvert` client class to use your custom endpoint. To do this, set the endpoint parameter on `AWS.config.mediaconvert`.

3. Define your transcoding job.

Create a Node.js module with the file name `emc_createjob.js`. Create the JSON that defines the transcode job parameters. These parameters are detailed. You can use the AWS Elemental MediaConvert console to generate the JSON job specification by choosing your job settings, and then choosing `Show job JSON` at the bottom of the `Job` section. For sample job specifications, see `Sample Settings in JSON` (p. 34).

For your Node.js module, wrap the JSON job specification as follows:

```javascript
var params = {
  "JSON job specification here"
};
```
4. Create your transcoding job.

After creating the job parameters JSON, call the `createJob` method by creating a promise for invoking an `AWS.MediaConvert` service object, passing the parameters. Then handle the response in the promise callback. The ID of the job that is created is returned in the response data:

```javascript
// Create a promise on a MediaConvert object
var endpointPromise = new AWS.MediaConvert({ apiVersion: '2017-08-29'}).createJob(params).promise();

// Handle the promise's fulfilled/rejected status
endpointPromise.then(
    function (data) {
        console.log('Job created! ', data);
    },
    function (err) {
        console.log('Error', err);
    }
);
```

To run the example, type the following at the command line:

`node ec2_createjob.js`

**.NET**

To send requests using the AWS SDK for .NET:

```csharp
using System;
using Amazon.MediaConvert;
using Amazon.MediaConvert.Model;

namespace MediaConvertNET
{
    /* ----------------------------------------------
    * Permissions IAM user needs to run this example
    * ----------------------------------------------
    * {
    *   "Version": "2012-10-17",
    *   "Statement": [
    *     
    *     "Sid": "VisualEditor0",
    *     "Effect": "Allow",
    *     "Action": [
    *       "mediaconvert:DescribeEndpoints",
    *       "mediaconvert:CreateJob"
    *     ],
    *     "Resource": "*"
    *   }
    */

    /* --------------------------------------
    * JSON job settings used in this example
    * --------------------------------------
```
* {  
  "UserMetadata": {  
    "Customer": "Amazon"
  },  
  "Role": "Your AWS Elemental MediaConvert role ARN",
  "Settings": {  
    "OutputGroups": [  
      {  
        "Name": "File Group",
        "OutputGroupSettings": {  
          "Type": "FILE_GROUP_SETTINGS",
          "FileGroupSettings": {  
            "Destination": "s3://youroutputdestination"
          }
        },  
        "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": 90,
                  "Slices": 1,
                  "GopBReference": "DISABLED",
                  "SlowPal": "DISABLED",
                  "SpatialAdaptiveQuantization": "ENABLED",
                  "TemporalAdaptiveQuantization": "ENABLED",
                  "FlickerAdaptiveQuantization": "DISABLED",
                  "EntropyEncoding": "CABAC",
                  "Bitrate": 5000000,
                  "FramerateControl": "SPECIFIED",
                  "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": "SPECIFIED",
                  "NumberBFramesBetweenReferenceFrames": 2,
                  "RepeatPps": "DISABLED",
                  "FramerateNumerator": 30,
                  "FramerateDenominator": 1,
                  "ParNumerator": 1,
                  "ParDenominator": 1
                }
              },  
              "AfdSignaling": "NONE",
              "DropFrameTimecode": "ENABLED",
              "RespondToAfd": "NONE",
            }
          }  
        ]
      }
    ]
  }
}
"ColorMetadata": "INSERT",
"AudioDescriptions": [
{
   "AudioTypeControl": "FOLLOW_INPUT",
   "CodecSettings": {
   "Codec": "AAC",
   "AacSettings": {
   "AudioDescriptionBroadcasterMix": "NORMAL",
   "RateControlMode": "CBR",
   "CodecProfile": "LC",
   "CodingMode": "CODING_MODE_2_0",
   "RawFormat": "NONE",
   "SampleRate": 48000,
   "Specification": "MPEG4",
   "Bitrate": 64000
   },
   "LanguageCodeControl": "FOLLOW_INPUT",
   "AudioSourceName": "Audio Selector 1"
   }
   },
   "ContainerSettings": {
   "Container": "MP4",
   "Mp4Settings": {
   "CsigAtom": "INCLUDE",
   "FreeSpaceBox": "EXCLUDE",
   "MoovPlacement": "PROGRESSIVE_DOWNLOAD"
   },
   "NameModifier": "_1"
   }
   }
   ],
"AdAvailOffset": 0,
"Inputs": [
{
   "AudioSelectors": {
   "Audio Selector 1": {
   "Offset": 0,
   "DefaultSelection": "NOT_DEFAULT",
   "ProgramSelection": 1,
   "SelectorType": "TRACK",
   "Tracks": [1]
   }
   },
   "VideoSelector": {
   "ColorSpace": "FOLLOW"
   },
   "FilterEnable": "AUTO",
   "PsiControl": "USE_PSI",
   "FilterStrength": 0,
   "DeblockFilter": "DISABLED",
   "DenoiseFilter": "DISABLED",
   "TimecodeSource": "EMBEDDED",
   "FileInput": "s3://yourinputfile"
   }
   ],
   "TimecodeConfig": {
   "Source": "EMBEDDED"
   }
   }
class Program
{
    static void Main(string[] args)
    {
        String mediaConvertRole = "Your AWS Elemental MediaConvert role ARN";
        String fileInput = "s3://yourinputfile";
        String fileOutput = "s3://youroutputdestination";
        // Once you know what your customer endpoint is, set it here
        String mediaConvertEndpoint = "";

        // If we do not have our customer-specific endpoint
        if (String.IsNullOrEmpty(mediaConvertEndpoint))
        {
            // Obtain the customer-specific MediaConvert endpoint
            AmazonMediaConvertClient client = new
                AmazonMediaConvertClient(Amazon.RegionEndpoint.USWest2);
            DescribeEndpointsRequest describeRequest = new
                DescribeEndpointsRequest();
            DescribeEndpointsResponse describeResponse =
                client.DescribeEndpoints(describeRequest);
            mediaConvertEndpoint = describeResponse.Endpoints[0].Url;
        }

        // Since we have a service url for MediaConvert, we do not
        // need to set RegionEndpoint. If we do, the ServiceURL will
        // be overwritten
        AmazonMediaConvertConfig mcConfig = new AmazonMediaConvertConfig
        {
            ServiceURL = mediaConvertEndpoint,
        };

        AmazonMediaConvertClient mcClient = new
            AmazonMediaConvertClient(mcConfig);

        CreateJobRequest createJobRequest = new CreateJobRequest();
        createJobRequest.Role = mediaConvertRole;
        createJobRequest.UserMetadata.Add("Customer", "Amazon");

        #region Create job settings
        JobSettings jobSettings = new JobSettings();
        jobSettings.AdAvailOffset = 0;
        jobSettings.TimecodeConfig = new TimecodeConfig();
        jobSettings.TimecodeConfig.Source = TimecodeSource.EMBEDDED;
        createJobRequest.Settings = jobSettings;
        #endregion

        #region OutputGroup
        OutputGroup ofg = new OutputGroup();
        ofg.Name = "File Group";
        ofg.OutputGroupSettings = new OutputGroupSettings();
        ofg.OutputGroupSettings.Type = OutputGroupType.FILE_GROUP_SETTINGS;

        Output output = new Output();
        output.NameModifier = "_1";
        #endregion

        VideoDescription vdes = new VideoDescription();
        output.VideoDescription = vdes;
        vdes.ScalingBehavior = ScalingBehavior.DEFAULT;
        vdes.TimecodeInsertion = VideoTimecodeInsertion.DISABLED;
        vdes.AntiAlias = AntiAlias.ENABLED;
        vdes.Sharpness = 50;
        vdes.AfdSignaling = AfdSignaling.NONE;
    }
}
vdes.DropFrameTimecode = DropFrameTimecode.ENABLED;
vdes.RespondToAfd = RespondToAfd.NONE;
vdes.ColorMetadata = ColorMetadata.INSERT;
vdes.CodecSettings = new VideoCodecSettings();
H264Settings h264 = new H264Settings();
h264.InterlaceMode = H264InterlaceMode.PROGRESSIVE;
h264.NumberReferenceFrames = 3;
h264.Syntax = H264Syntax.DEFAULT;
h264.Softness = 0;
h264.GopClosedCadence = 1;
h264.GopSize = 90;
h264.Slices = 1;
h264.GopBReference = H264GopBReference.DISABLED;
h264.SlowPal = H264SlowPal.DISABLED;
h264.SpatialAdaptiveQuantization = H264SpatialAdaptiveQuantization.ENABLED;
h264.TemporalAdaptiveQuantization = H264TemporalAdaptiveQuantization.ENABLED;
h264.FlickerAdaptiveQuantization = H264FlickerAdaptiveQuantization.DISABLED;
h264.EntropyEncoding = H264EntropyEncoding.CABAC;
h264.Bitrate = 5000000;
h264.FramerateControl = H264FramerateControl.SPECIFIED;
h264.RateControlMode = H264RateControlMode.CBR;
h264.CodecProfile = H264CodecProfile.MAIN;
h264.Telecine = H264Telecine.NONE;
h264.MinIInterval = 0;
h264.AdaptiveQuantization = H264AdaptiveQuantization.HIGH;
h264.CodecLevel = H264CodecLevel.AUTO;
h264.FieldEncoding = H264FieldEncoding.PAFF;
h264.SceneChangeDetect = H264SceneChangeDetect.ENABLED;
h264.QualityTuningLevel = H264QualityTuningLevel.SINGLE_PASS;
h264.FramerateConversionAlgorithm = H264FramerateConversionAlgorithm.DUPLICATE_DROP;
h264.UnregisteredSeiTimecode = H264UnregisteredSeiTimecode.DISABLED;
h264.GopSizeUnits = H264GopSizeUnits.FRAMES;
h264.ParControl = H264ParControl.SPECIFIED;
h264.NumberBFramesBetweenReferenceFrames = 2;
h264.RepeatPps = H264RepeatPps.DISABLED;
h264.FramerateNumerator = 30;
h264.FramerateDenominator = 1;
h264.ParNumerator = 1;
h264.ParDenominator = 1;
output.VideoDescription.CodecSettings.H264Settings = h264;
#region AudioDescription
AudioDescription ades = new AudioDescription();
ades.LanguageCodeControl = AudioLanguageCodeControl.FOLLOW_INPUT;
// This name matches one specified in the Inputs below
ades.AudioSourceName = "Audio Selector 1";
ades.CodecSettings = new AudioCodecSettings();
ades.CodecSettings.Codec = AudioCodec.AAC;
AacSettings aac = new AacSettings();
aac.AudioDescriptionBroadcasterMix = AacAudioDescriptionBroadcasterMix.NORMAL;
aac.RateControlMode = AacRateControlMode.CBR;
aac.CodecProfile = AacCodecProfile.LC;
aac.CodingMode = AacCodingMode.CODING_MODE_2_0;
aac.RawFormat = AacRawFormat.NONE;
aac.SampleRate = 48000;
aac.Specification = AacSpecification.MPEG4;
aac.Bitrate = 64000;
ades.CodecSettings.AacSettings = aac;
output.AudioDescriptions.Add(ades);
#endregion AudioDescription

#region Mp4 Container
output.ContainerSettings = new ContainerSettings();
output.ContainerSettings.Container = ContainerType.MP4;
Mp4Settings mp4 = new Mp4Settings();
mp4.CslgAtom = Mp4CslgAtom.INCLUDE;
mp4.FreeSpaceBox = Mp4FreeSpaceBox.EXCLUDE;
mp4.MoovPlacement = Mp4MoovPlacement.PROGRESSIVE_DOWNLOAD;
output.ContainerSettings.Mp4Settings = mp4;
#endregion Mp4 Container

ofg.Outputs.Add(output);
createJobRequest.Settings.OutputGroups.Add(ofg);
#endregion OutputGroup

#region Input
Input input = new Input();
input.FilterEnable = InputFilterEnable.AUTO;
input.PsiControl = InputPsiControl.USE_PSI;
input.FilterStrength = 0;
input.DeblockFilter = InputDeblockFilter.DISABLED;
input.DenoiseFilter = InputDenoiseFilter.DISABLED;
input.TimecodeSource = InputTimecodeSource.EMBEDDED;
input.FileInput = fileInput;

AudioSelector audsel = new AudioSelector();
audsel.Offset = 0;
audsel.DefaultSelection = AudioDefaultSelection.NOT_DEFAULT;
audsel.ProgramSelection = 1;
audsel.SelectorType = AudioSelectorType.TRACK;
audsel.Tracks.Add(1);
input.AudioSelectors.Add("Audio Selector 1", audsel);

input.VideoSelector = new VideoSelector();
input.VideoSelector.ColorSpace = ColorSpace.FOLLOW;

createJobRequest.Settings.Inputs.Add(input);
#endregion Input

#region Create job settings
try
{
    CreateJobResponse createJobResponse = mcClient.CreateJob(createJobRequest);
    Console.WriteLine("Job Id: {0}", createJobResponse.Job.Id);
}
catch (BadRequestException bre)
{
    // If the endpoint was bad
    if (bre.Message.StartsWith("You must use the customer-"))
    {
        // The exception contains the correct endpoint; extract it
        mediaConvertEndpoint = bre.Message.Split(\"\")[1];
        // Code to retry query
    }
}
}
For more information about sending requests to an account endpoint, see the Overriding Endpoints in the AWS SDK for .NET post in the AWS Developer Blog.

**PHP**

After you use the DescribeEndpoints method to request an account-specific endpoint, send your requests to it as described in the Configuration Options > endpoint topic in the AWS SDK for PHP Developer Guide.

To send AWS Elemental MediaConvert requests using the AWS SDK for PHP:

1. Add the following import statements:

   ```php
   require 'vendor/autoload.php';
   use Aws\MediaConvert\MediaConvertClient;
   use Aws\Exception\AwsException;
   ```

2. Determine the region for your endpoint and create an AWSMediaConvert client object for it:

   ```php
   $client = new Aws\MediaConvert\MediaConvertClient(
       'profile' => 'default',
       'version' => '2017-08-29',
       'region'  => 'us-east-1'
   );
   ```

3. Call the describeEndpoints method to retrieve the endpoint and save the endpoint’s URL:

   ```php
   $URI = $client->getEndpoint() //Returns UriInterface
   $endpoints = $client->describeEndpoints(['NextToken' => '']);
   $token = $URI->getHost();
   try {
       $result = $client->describeEndpoints([]);
   } catch (AwsException $e) {
       // output error message if fails
       echo $e->getMessage();
       echo "\n";
   }
   $single_endpoint_url = $result['Endpoints'][0]['Url'];
   ```

4. Create another AWSMediaConvert client object using the region and the endpoint URL that you just retrieved:

   ```php
   $mediaConvertClient = new MediaConvertClient(
       'profile' => 'default',
       'version' => '2017-08-29',
       'region'  => 'us-east-1',
       'endpoint' => $single_endpoint_url
   );
   ```

   This is the client object to use for all of your job requests.

5. Create a job request and submit it:

   ```php
   try {
       $result = $mediaConvertClient -> createJob(
           'ClientRequestToken' => , //String Idempotency token for CreateJob operation.
           'JobTemplate' => , //String either a job template or the transcoding settings
   ```
Role => 'ARN',  //REQUIRED The IAM role you use for creating this job.
Settings => $jobSetting, //Settings is a JobSettings object that contains Settings parameters.
Queue => '', //Optional
});
}catch (AwsException $e) {
  // output error message if fails
  echo $e->getMessage();
  echo "\n";
}


Each job converts an input file into an output file or files. For more information, see the AWS Elemental MediaConvert User Guide at http://docs.aws.amazon.com/mediaconvert/latest/ug/.

Simple example: Job settings for AWS Elemental MediaConvert:

```
$jobSetting = [
  "OutputGroups" => [
    [
      "Name" => "File Group",
      "OutputGroupSettings" => [
        "Type" => "FILE_GROUP_SETTINGS",
        "FileGroupSettings" => [
          "Destination" => "s3://testbucket/output"
        ],
      ],
    ],
    "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" => 90,
              "Slice" => 1,
              "GopReference" => "DISABLED",
              "SlowPal" => "DISABLED",
              "SpatialAdaptiveQuantization" => "ENABLED",
              "TemporalAdaptiveQuantization" => "ENABLED",
              "FlickerAdaptiveQuantization" => "DISABLED",
              "EntropyEncoding" => "CABAC",
              "Bitrate" => 5000000,
              "FramerateControl" => "SPECIFIED",
              "RateControlMode" => "CBR",
              "CodecProfile" => "MAIN",
              "Telecine" => "NONE",
              "MinInterval" => 0,
              "AdaptiveQuantization" => "HIGH",
              "CodecLevel" => "AUTO",
              "FieldEncoding" => "PAFF",
              "SceneChangeDetect" => "ENABLED",
            ],
          ],
        ],
      ],
    ],
  ],
];
```
"QualityTuningLevel" => "SINGLE_PASS",
"FramerateConversionAlgorithm" => "DUPLICATE_DROP",
"UnregisteredSeiTimecode" => "DISABLED",
"GopSizeUnits" => "FRAMES",
"ParControl" => "SPECIFIED",
"NumberBFramesBetweenReferenceFrames" => 2,
"RepeatPps" => "DISABLED",
"FramerateNumerator" => 30,
"FramerateDenominator" => 1,
"ParNumerator" => 1,
"ParDenominator" => 1,

"AfdSignaling" => "NONE",
"DropFrameTimecode" => "ENABLED",
"RespondToAfd" => "NONE",
"ColorMetadata" => "INSERT",

"AudioDescriptions" => [
    
    "AudioTypeControl" => "FOLLOW_INPUT",
    "CodecSettings" => [
        "Codec" => "AAC",
        "AacSettings" => [
            "AudioDescriptionBroadcasterMix" => "NORMAL",
            "RateControlMode" => "CBR",
            "CodecProfile" => "LC",
            "CodingMode" => "CODING_MODE_2_0",
            "RawFormat" => "NONE",
            "SampleRate" => 48000,
            "Specification" => "MPEG4",
            "Bitrate" => 64000
        ],
        "LanguageCodeControl" => "FOLLOW_INPUT",
        "AudioSourceName" => "Audio Selector 1"
    ],

    "ContainerSettings" => [
        "Container" => "MP4",
        "Mp4Settings" => [
            "CslgAtom" => "INCLUDE",
            "FreeSpaceBox" => "EXCLUDE",
            "MoovPlacement" => "PROGRESSIVE_DOWNLOAD"
        ],
        "NameModifier" => "_1"
    ]
],

"AdAvailOffset" => 0,
"Inputs" => [
    [
        "AudioSelectors" => [
            "Audio Selector 1" => [
                "Offset" => 0,
                "DefaultSelection" => "NOT_DEFAULT",
                "ProgramSelection" => 1,
                "SelectorType" => "TRACK",
                "Tracks" => [1]
            ]
        ],
        "VideoSelector" => [18]
Complex example: Job settings for AWS Elemental MediaConvert:

```
"UserMetadata" => {
  "Customer" => "Amazon",
  "Role" => "arn:aws:iam::111122223333:role/MediaConvertRole",
"Settings" => {
  "OutputGroups" => {
    "CustomName" => "",
    "Name" => "DASH ISO",
    "Outputs" => {
      "ContainerSettings" => {
        "Container" => "MPD"
      },
      "VideoDescription" => {
        "Width" => 1080,
        "ScalingBehavior" => "DEFAULT",
        "Height" => 720,
        "TimecodeInsertion" => "DISABLED",
        "AntiAlias" => "ENABLED",
        "Sharpness" => 50,
        "CodecSettings" => {
          "Codec" => "H_264",
          "H264Settings" => {
            "InterlaceMode" => "PROGRESSIVE",
            "NumberReferenceFrames" => 3,
            "Syntax" => "DEFAULT",
            "Softness" => 0,
            "GopClosedCadence" => 1,
            "GopSize" => 30,
            "Slices" => 1,
            "GopBReference" => "DISABLED",
            "SlowPal" => "DISABLED",
            "SpatialAdaptiveQuantization" => "ENABLED",
            "TemporalAdaptiveQuantization" => "ENABLED",
            "FlickerAdaptiveQuantization" => "DISABLED",
            "EntropyEncoding" => "CABAC",
            "Bitrate" => 5000000,
            "FramerateControl" => "INITIALIZE_FROM_SOURCE",
            "RateControlMode" => "CBR",
            "CodecProfile" => "MAIN",
            "Telecine" => "NONE",
            "MinInterval" => 0,
            "AdaptiveQuantization" => "HIGH",
          }
        }
      }
    }
  }
}
```

"CodecLevel" => "AUTO",
"FieldEncoding" => "PAFF",
"SceneChangeDetect" => "ENABLED",
"QualityTuningLevel" => "SINGLE_PASS",
"FramerateConversionAlgorithm" => "DUPLICATE_DROP",
"UnregisteredScteTimecode" => "DISABLED",
"GopSizeUnits" => "FRAMES",
"ParControl" => "INITIALIZE_FROM_SOURCE",
"NumberBFramesBetweenReferenceFrames" => 2,
"RepeatPps" => "DISABLED",
"HrdBufferSize" => 10000000
],
"AfdSignaling" => "NONE",
"DropFrameTimecode" => "ENABLED",
"RespondToAfd" => "NONE",
"ColorMetadata" => "INSERT"
],
"NameModifier" => "-1080"
],
[ "ContainerSettings" => [
  "Container" => "MPD"
],
"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"
  ],
  "NameModifier" => "-audio"
],
"OutputGroupSettings" => [
  "Type" => "DASH_ISO_GROUP_SETTINGS",
  "DashIsoGroupSettings" => [
    "SegmentLength" => 30,
    "Destination" => "s3 => //testbucket/drm/10/dash-drm/master",
    "Encryption" => [
      "SpekeKeyProvider" => [
        "ResourceId" => "drm-test-1",
        "SystemIds" => [
          "edef8ba9-79d6-4ace-a3c8-27dcd51d21ed"
        ],
        "Url" => "https => //abcdefg123.execute-api.us-west-2.amazonaws.com/live/speke/v1.0/copyProtection"
      ],
      "FragmentLength" => 2,
      "SegmentControl" => "SINGLE_FILE",
      "HbbtvCompliance" => "NONE"
    ]
  ]
]
"Name" => "Apple HLS",
"OutputGroupSettings" => [ 
  "Type" => "HLS_GROUP_SETTINGS",
  "HlsGroupSettings" => [ 
    "ManifestDurationFormat" => "INTEGER",
    "SegmentLength" => 10,
    "TimedMetadataId3Period" => 10,
    "CaptionLanguageSetting" => "OMIT",
    "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",
    "Destination" => "s3 => //testbucket/hls/no-drm/master"
  ],
  "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" => 90,
          "Slices" => 1,
          "Gop8Reference" => "DISABLED",
          "SlowPal" => "DISABLED",
          "SpatialAdaptiveQuantization" => "ENABLED",
          "TemporalAdaptiveQuantization" => "ENABLED",
          "FlickerAdaptiveQuantization" => "DISABLED",
          "EntropyEncoding" => "CABAC",
          "Bitrate" => 5000000,
          "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"
],
"AudioDescriptions" => [
  [
    "AudioTypeControl" => "FOLLOW_INPUT",
    "CodecSettings" => [
      "Codec" => "AAC",
      "AacSettings" => [
        "AudioDescriptionBroadcasterMix" => "NORMAL",
        "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",
      "NielsenId3" => "NONE",
      "TimedMetadata" => "NONE",
      "VideoPid" => 481,
      "AudioPids" => [482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492]
    ]
  ],
  "NameModifier" => "_v1"
],
[ "AudioDescriptions" => [
  [ "AudioTypeControl" => "FOLLOW_INPUT",

"CodecSettings" => [
  "Codec" => "AAC",
  "AacSettings" => [
    "AudioDescriptionBroadcasterMix" => "NORMAL",
    "RateControlMode" => "CBR",
    "CodecProfile" => "LC",
    "CodingMode" => "CODING_MODE_2_0",
    "RawFormat" => "NONE",
    "SampleRate" => 48000,
    "Specification" => "MPEG4",
    "Bitrate" => 64000
  ]
],
"LanguageCodeControl" => "FOLLOW_INPUT",
"AudioSourceName" => "Audio Selector 1",
"StreamName" => "English",
"LanguageCode" => "ENG"
],
"OutputSettings" => [
  "HlsSettings" => [
    "AudioGroupId" => "program_audio",
    "AudioRenditionSets" => "program_audio",
    "IFrameOnlyManifest" => "EXCLUDE",
    "AudioTrackType" => "ALTERNATE_AUDIO_AUTO_SELECT_DEFAULT"
  ]
],
"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]
  ],
  "NameModifier" => "_a1"
],
[ "AudioDescriptions" => [
  ["AudioTypeControl" => "FOLLOW_INPUT",
    "CodecSettings" => [
      "Codec" => "AAC",
      "AacSettings" => [
        "AudioDescriptionBroadcasterMix" => "NORMAL",
        "RateControlMode" => "CBR",
        "CodecProfile" => "LC",
        "CodingMode" => "CODING_MODE_2_0",
        "RawFormat" => "NONE",
        "SampleRate" => 48000,
        "Specification" => "MPEG4",
        "Bitrate" => 64000
      ]
    ]
  ]
"RateControlMode" => "CBR",
"CodecProfile" => "LC",
"CodingMode" => "CODING_MODE_2_0",
"RawFormat" => "NONE",
"SampleRate" => 48000,
"Specification" => "MPEG4",
"Bitrate" => 64000 ]
],
"LanguageCodeControl" => "FOLLOW_INPUT",
"AudioSourceName" => "Audio Selector 2",
"StreamName" => "Spanish",
"LanguageCode" => "SPA"
],
"OutputSettings" => [
"HlsSettings" => [
"AudioGroupId" => "program_audio",
"AudioRenditionSets" => "program_audio",
"IFrameOnlyManifest" => "EXCLUDE",
"AudioTrackType" => "ALTERNATE_AUDIO_AUTO_SELECT"
]
],
"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]
]
],
"NameModifier" => "_a2"
]
],
"AdAvailOffset" => 0,
"Inputs" => [
"AudioSelectors" => [
"Audio Selector 1" => [
"Offset" => 0,
"DefaultSelection" => "DEFAULT",
"ProgramSelection" => 1,
"SelectorType" => "TRACK",
24
Python

To send requests using the AWS SDK for Python (Boto):

1. Use the `describe_endpoints` method to request an account-specific endpoint.

   **Important**
   Make the `DescribeEndpoints` call only once in your application. Don't use `DescribeEndpoints` to create your AWS client each time that you make a request to AWS Elemental MediaConvert. Otherwise, you will reach the throttle maximum on the public API endpoint.

2. Specify the transcoding settings for your job in a JSON file.

   You can use the AWS Elemental MediaConvert console to generate the JSON job specification by choosing your job settings, and then choosing `Show job JSON` at the bottom of the Job section. For sample job specifications, see `Sample Settings in JSON` (p. 34).

3. Perform these steps in your Python code, as demonstrated in the example that follows these steps:
   a. Create the client with your account-specific endpoint specified.
   b. Load your settings JSON file.
   c. Create the transcoding job using `create_job`.

```python
import json
import boto3

# Create MediaConvert client
mediaconvert_client = boto3.client('mediaconvert', endpoint_url='https://abcd1234.mediaconvert.us-west-2.amazonaws.com')
```
# Load job.json from disk and store as Python object: job_object
with open("job.json", "r") as jsonfile:
    job_object = json.load(jsonfile)

# Create MediaConvert job by unpacking the arguments from job_object. The job
object contains the required parameters
# for create_job. Pass these to create_job using Python's ** argument unpacking
syntax.
mediaconvert_client.create_job(**job_object)

Ruby

After you use the DescribeEndpointsRequest method to request an account-specific endpoint,
send your requests to it as described in the Setting a Nonstandard Endpoint topic in the AWS
SDK for Ruby Developer Guide.

Important
Make the DescribeEndpoints call only once in your application. Don't use
DescribeEndpoints to create your AWS client each time that you make a request to
AWS Elemental MediaConvert. Otherwise, you will reach the throttle maximum on the
public API endpoint.

Tools for Powershell

After you use the Get-EMCEndpoint Cmdlet to request an account-specific endpoint, send your
requests to it as described in the Specifying a Custom or Nonstandard Endpoint topic in the AWS
Tools for PowerShell User Guide. In brief, you specify your account-specific endpoint as a URL by
adding the -EndpointUrl common parameter to your AWS Tools for PowerShell command.

In this example, replace https://abcd1234.mediaconvert.us-west-1.amazonaws.com
with the endpoint that you get back from your Get-EMCEndpoint Cmdlet request:

AWS-PowerShellCmdlet -EndpointUrl "https://abcd1234.mediaconvert.us-
west-1.amazonaws.com." -Other -Parameters

Important
Make the DescribeEndpoints call only once in your application. Don't use
DescribeEndpoints to create your AWS client each time that you make a request to
AWS Elemental MediaConvert. Otherwise, you will reach the throttle maximum on the
public API endpoint.
Getting Started with AWS Elemental MediaConvert Using the API

Important
Unlike most AWS services, AWS Elemental MediaConvert requires that you send your requests to an endpoint that is specific to your account. Use the following steps to get going.

1. **Set up permissions:**
   - **Permissions that the AWS Elemental MediaConvert service can assume on your behalf.** These allow access to your S3 buckets and API Gateway. For instructions, see the Set Up IAM Permissions topic of the AWS Elemental MediaConvert User Guide.
   - **Your Signature Version 4 authentication for the requests you send to AWS.** When you use the AWS Command Line Interface or one of the AWS SDKs, these tools automatically sign the requests for you with the access key and secret key that you specify in your client configuration.

2. **Set up S3 file locations.** The service reads your input files from and saves your output files to Amazon S3 buckets. For instructions on creating these buckets, see the Create Storage for Files topic of the AWS Elemental MediaConvert User Guide.

3. **Request your account endpoint.** Send a POST request with an empty body to the following endpoint, substituting in your region. Usually, this is the region where you store your input and output files. Note that this region must match the region you set up in your client configuration.

   ```
   https://mediaconvert.<region>.amazonaws.com/2017-08-29/endpoints
   ```

   For example, for the region `ap-southeast-2`, the endpoint is the following:

   ```
   https://mediaconvert.ap-southeast-2.amazonaws.com/2017-08-29/endpoints
   ```

   For a full list of AWS Elemental MediaConvert public endpoints, see AWS Elemental MediaConvert Regions and Endpoints.

   The service returns an endpoint similar to “https://abcd1234.mediaconvert.us-west-1.amazonaws.com”, where the first eight digits after “https://” are an alphanumeric combination unique to your account and the region matches the regional endpoint that you send the request to. This is the endpoint to use for your transcoding requests.

   Important
   If you request your account endpoint programatically, only do so once in your application. Don’t make a request to the public endpoint each time you make a request to AWS Elemental MediaConvert. Otherwise, you will hit the throttle maximum on the public API endpoint.

4. **Send your transcoding requests.**

   Using the account endpoint you received from your POST request, send your requests to manage transcoding jobs, queues, job templates, and presets. For general information about how these resources work, see the AWS Elemental MediaConvert User Guide. For high level information about each resource, including information about each transcoding setting, see the Resources portion of this guide.
If you use Postman to send your requests, you can optionally copy and import the preconfigured Postman Collection Files (p. 28).

Postman Collection Files

Import these collection files into Postman to access AWS Elemental MediaConvert via the REST API.

Topics

- GET Collection (p. 28)
- POST Collection (p. 30)

GET Collection

```
{
  "owner": "2332976",
  "lastUpdatedBy": "2332976",
  "lastRevision": 1921667904,
  "team": null,
  "id": "87fac2df-dd0f-b54a-b1f9-5b138cb4147f",
  "name": "EMF Get",
  "description": "EMF Get Template",
  "folders_order": [],
  "order": [
    "bc671df5-4a85-54b6-f137-19cb70516fd2",
    "85318a0b-c490-3718-62eb-2a737de83af0",
    "1fd40def-ca4b-1842-c99a-778f62269010"
  ],
  "folders": [],
  "hasRequests": true,
  "requests": [
    {
      "id": "1fd40def-ca4b-1842-c99a-778f62269010",
      "headers": "Content-Type: application/json",
      "headerData": [
        {
          "key": "Content-Type",
          "value": "application/json",
          "description": "",
          "enabled": true
        }
      ],
      "url": "https://<custom-account-id>.mediaconvert.<region>.amazonaws.com/2017-08-29/queues",
      "folder": null,
      "queryParams": [],
      "preRequestScript": null,
      "pathVariables": {},
      "pathVariableData": [],
      "method": "GET",
      "data": null,
      "dataMode": "params",
      "tests": null,
      "currentHelper": "awsSigV4",
      "helperAttributes": {
        "accessKey": "AccessKey",
```
"secretKey": "SecretKey",
"region": "supported-region",
"service": "mediaconvert",
"saveToRequest": true
},
"time": 1513791262493,
"name": "GET List Queue ",
"description": "",
"collectionId": "87fac2df-dd0f-b54a-b1f9-5b138cb4147f",
"responses": []
},
{
"id": "85318a0b-c490-3718-62eb-2a737de83af0",
"headers": "Content-Type: application/json",
"headerData": [
{
 "key": "Content-Type",
 "value": "application/json",
 "description": "",
 "enabled": true
}
],
"url": "https://<custom-account-id>.mediaconvert.<region>.amazonaws.com/2017-08-29/
 queues/<QUEUE-NAME-HERE>",
"folder": null,
"queryParams": [],
"preRequestScript": null,
"pathVariables": {},
"pathVariableData": [],
"method": "GET",
"data": null,
"dataMode": "params",
"tests": null,
"currentHelper": "awsSigV4",
"helperAttributes": {
 "accessKey": "AccessKey",
 "secretKey": "SecretKey",
 "region": "supported-region",
 "service": "mediaconvert",
 "saveToRequest": true
 },
"time": 1507243078514,
"name": "GET Queue Details",
"description": "",
"collectionId": "87fac2df-dd0f-b54a-b1f9-5b138cb4147f",
"responses": []
},
{
"id": "bc671df5-4a85-54b6-f137-19cb70516fd2",
"headers": "Content-Type: application/json",
"headerData": [
{
 "key": "Content-Type",
 "value": "application/json",
 "description": "",
 "enabled": true
}
],
"url": "https://<custom-account-id>.mediaconvert.<region>.amazonaws.com/2017-08-29/jobs/
<job-id>",
"folder": null,
"queryParams": [],
"preRequestScript": null,
"pathVariables": {},
"pathVariableData": [],
"method": "GET"
POST Collection

```json
{
  "id": "a1be92f5-37d5-aaf0-06bb-14090d825c62",
  "name": "AWS Elemental MediaConvert POST",
  "description": "POST Template",
  "order": [],
  "folders": [],
  "timestamp": 0,
  "owner": "2332976",
  "public": false,
  "requests": [
    {
      "id": "0fd3c4a5-fa08-2dbc-1f0a-955942664858",
      "headers": "Content-Type: application/json\n",
      "headerData": [
        {
          "key": "Content-Type",
          "value": "application/json",
          "description": "",
          "enabled": true
        }
      ],
      "url": "https://<custom-account-id>.mediaconvert.<region>.amazonaws.com/2017-08-29/",
      "folder": null,
      "queryParams": [
        {
          "key": "AWS_Region",
          "value": "eu-west-1",
          "equals": false,
          "description": "",
          "enabled": false
        },
        {
          "key": "AWS_Access_Key",
```
"value": "KEY",
"equals": false,
"description": "",
"enabled": false
},

{  
  "key": "AWS_Secret_Key",
  "value": "KEY",
  "equals": false,
  "description": "",
  "enabled": false
}
],
"preRequestScript": "",
"pathVariables": {},
"pathVariableData": [],
"method": "POST",
"data": [],
"dataMode": "raw",
"tests": "",
"currentHelper": "awsSigV4",
"helperAttributes": {
  "accessKey": "AccessKey",
  "secretKey": "SecretKey",
  "region": "supported-region",
  "service": "mediaconvert",
  "saveToRequest": true
},
"time": 1510272274641,
"name": "Post MP4 Job",
"description": "",
"collectionId": "a1be92f5-37d5-aaf0-06bb-14090d825c62",
"responses": [],
"rawModeData": "{\"userMetadata\": {},\"role\": \"ROLE ARN HERE\",\n\"settings\": {\"outputGroups\": [\n  \"container\": \"MP4\",
  \"cslgAtom\": \"INCLUDE\",
  \"freeSpaceBox\": \"EXCLUDE\",
  \"moovPlacement\": \"PROGRESSIVE_DOWNLOAD\",
  \"videoDescription\": {\n    \"scalingBehavior\": \"DEFAULT\",
    \"antiAlias\": \"ENABLED\",
    \"timecodeDescription\": \"DISABLED\",
    \"timecodeInsertion\": \"DISABLED\",
    \"renderStream\": \"DISABLED\",
    \"sharpness\": 50,
    \"codecSettings\": {\n      \"h264Settings\": {\n        \"interlaceMode\": \"PROGRESSIVE\",
        \"numberReferenceFrames\": 1,
        \"syntax\": \"DEFAULT\",
        \"gopClosedCadence\": 1,
        \"gopSize\": 90,
        \"numBFramesBetweenReferenceFrames\": 2,
        \"frameRateControl\": \"INITIALIZE_FROM_SOURCE\",
        \"telecine\": \"NONE\",
        \"adaptiveQuantization\": \"MEDIUM\",
        \"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\",
        \"audioTypeControl\": \"FOLLOW_INPUT\",
        \"codecSettings\": {\n          \"defaultCodec\": \"H_264\",
          \"interlaceMode\": \"PROGRESSIVE\",
          \"numberReferenceFrames\": 1,
          \"syntax\": \"DEFAULT\",
          \"gopClosedCadence\": 1,
          \"gopSize\": 90,
          \"numBFramesBetweenReferenceFrames\": 2,
          \"frameRateControl\": \"INITIALIZE_FROM_SOURCE\",
          \"telecine\": \"NONE\",
          \"adaptiveQuantization\": \"MEDIUM\",
          \"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\",
          \"audioTypeControl\": \"FOLLOW_INPUT\",
          \"codecSettings\": {}\n      }\n    }\n  }\n}"}
}
"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"
},
"inputs": [
  {
    "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": "EMBEDDED",
    "fileInput": "s3://bucket/file.mp4"
  }
],
"adAvailOffset": 0,
"inputs": [
  {
    "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": "EMBEDDED",
    "fileInput": "s3://bucket/file.mp4"
  }
],
"outputGroupSettings": {
  "type": "FILE_GROUP_SETTINGS",
  "fileGroupSettings": {
    "destination": "s3://test/test"
  }
},
"adAvailOffset": 0,
"inputs": [
  {
    "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": "EMBEDDED",
    "fileInput": "s3://bucket/file.mp4"
  }
],
"adAvailOffset": 0,
"inputs": [
  {
    "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": "EMBEDDED",
    "fileInput": "s3://bucket/file.mp4"
  }
],
"adAvailOffset": 0,
"inputs": [
  {
    "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": "EMBEDDED",
    "fileInput": "s3://bucket/file.mp4"
  }
]
"region": "Supported Region",
"service": "mediaconvert",
"saveToRequest": true
},
"time": 1510272153358,
"name": "POST Request Account Endpoint",
"description": "",
"collectionId": "a1be92f5-37d5-aaf0-06bb-14090d825c62",
"responses": []
]}
"rawModeData": ""
]}
}
Sample Job Settings Specifications in JSON

These sample AWS Elemental MediaConvert jobs are specified in JSON. When you use the Python SDK or the AWS CLI, you pass in your job settings directly as a JSON object. When you use the other SDKs, translate the job settings according to the SDK documentation.

Simple Example

This example JSON job settings specification describes a job that creates a single file output.

```json
{
    "UserMetadata": {
        "Customer": "Amazon"
    },
    "Role": "arn:aws:iam::505474453218:role/EMFRoleSPNames",
    "Settings": {
        "OutputGroups": [
            {
                "Name": "File Group",
                "OutputGroupSettings": {
                    "Type": "FILE_GROUP_SETTINGS",
                    "FileGroupSettings": {
                        "Destination": "s3://testbucket/output"
                    }
                }
            }
        ],
        "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": 90,
                            "Slices": 1,
                            "GopBReference": "DISABLED",
                            "SlowPal": "DISABLED",
                            "SpatialAdaptiveQuantization": "ENABLED",
                            "TemporalAdaptiveQuantization": "ENABLED",
                            "FlickerAdaptiveQuantization": "DISABLED",
                            "EntropyEncoding": "CABAC",
                            "Bitrate": 5000000,
                            "FramerateControl": "SPECIFIED",
                            "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": "SPECIFIED",
"NumberBFramesBetweenReferenceFrames": 2,
"RepeatPps": "DISABLED",
"FramerateNumerator": 30,
"FramerateDenominator": 1,
"ParNumerator": 1,
"ParDenominator": 1
}

"AfdSignaling": "NONE",
"DropFrameTimecode": "ENABLED",
"RespondToFad": "NONE",
"ColorMetadata": "INSERT"
},
"AudioDescriptions": [
{
"AudioTypeControl": "FOLLOW_INPUT",
"CodecSettings": {
"Codec": "AAC",
"AacSettings": {
"AudioDescriptionBroadcasterMix": "NORMAL",
"RateControlMode": "CBR",
"CodecProfile": "LC",
"CodingMode": "CODING_MODE_2_0",
"RawFormat": "NONE",
"SampleRate": 48000,
"Specification": "MPEG4",
"Bitrate": 64000
}
},
"LanguageCodeControl": "FOLLOW_INPUT",
"AudioSourceName": "Audio Selector 1"
}
],
"ContainerSettings": {
"Container": "MP4",
"Mp4Settings": {
"CsrlAtom": "INCLUDE",
"FreeSpaceBox": "EXCLUDE",
"MoovPlacement": "PROGRESSIVE_DOWNLOAD"
}
},
"NameModifier": "_1"
}
],
"AdAvailOffset": 0,
"Inputs": [
{
"AudioSelectors": {
"Audio Selector 1": {
"Offset": 0,
"DefaultSelection": "NOT_DEFAULT",
"ProgramSelection": 1,
"SelectorType": "TRACK",
"Tracks": [
Complex Example

This example JSON job settings specification describes a job that creates output in multiple packages for viewing on different device types and uses encryption for digital rights management.

```json
{
    "UserMetadata": { "Customer": "Amazon" },
    "Role": "arn:aws:iam::111122223333:role/MediaConvertRole",
}
```
"SpatialAdaptiveQuantization": "ENABLED",
"TemporalAdaptiveQuantization": "ENABLED",
"FlickerAdaptiveQuantization": "DISABLED",
"EntropyEncoding": "CABAC",
"Bitrate": 5000000,
"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",
"HrdBufferSize": 10000000
},

"AfdSignaling": "NONE",
"DropFrameTimecode": "ENABLED",
"RespondToAfd": "NONE",
"ColorMetadata": "INSERT"
},
"NameModifier": "-1080"
},
{
"ContainerSettings": {
"Container": "MPD"
},
"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"
}
],
"NameModifier": "-audio"
}

"OutputGroupSettings": {
"Type": "DASH_ISO_GROUP_SETTINGS",
"DashIsoGroupSettings": {
"SegmentLength": 30,
"Destination": "s3://testbucket/drm/10/dash-drm/master",
"Encryption": {
"SpekeKeyProvider": {
"ResourceId": "drm-test-1",
"Resourceid": "drm-test-1",
"Username": "testuser",
"Password": "testpass"
}
"SystemIds": ["edef8ba9-79d6-4ace-a3c8-27dcd51d21ed"],
"Url": "https://abcdefg123.execute-api.us-west-2.amazonaws.com/live/speke/v1.0/copyProtection"},
},
"FragmentLength": 2,
"SegmentControl": "SINGLE_FILE",
"HbbtvCompliance": "NONE"
}
],
"Name": "Apple HLS",
"OutputGroupSettings": {
"Type": "HLS_GROUP_SETTINGS",
"HlsGroupSettings": {
"ManifestDurationFormat": "INTEGER",
"SegmentLength": 10,
"TimedMetadataId3Period": 10,
"CaptionLanguageSetting": "OMIT",
"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",
"Destination": "s3://testbucket/hls/no-drm/master"
}
},
"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": 90,
"Slices": 1,
"GopBReference": "DISABLED",
"SlowPal": "DISABLED",
"SpatialAdaptiveQuantization": "ENABLED",
"TemporalAdaptiveQuantization": "ENABLED",
"FlickerAdaptiveQuantization": "DISABLED",
"EntropyEncoding": "CABAC",
"Bitrate": 5000000,
"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"
},
"AudioDescriptions": [
{
"AudioTypeControl": "FOLLOW_INPUT",
"CodecSettings": {
"Codec": "AAC",
"AacSettings": {
"AudioDescriptionBroadcasterMix": "NORMAL",
"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,
"PmtInterval": 0,
"PmtInterval": 0,
"Scte35Source": "NONE",
"NielsenId3": "NONE",
"TimedMetadata": "NONE",
"VideoPid": 481,
"AudioPids": [482, 483, 484, 485, 486, 487, 488, 489, 490, 491]"}


```
{
  "NameModifier": "_v1",
  "AudioDescriptions": [
    {
      "AudioTypeControl": "FOLLOW_INPUT",
      "CodecSettings": {
        "Codec": "AAC",
        "AacSettings": {
          "AudioDescriptionBroadcasterMix": "NORMAL",
          "RateControlMode": "CBR",
          "CodecProfile": "LC",
          "CodingMode": "CODING_MODE_2_0",
          "RawFormat": "NONE",
          "SampleRate": 48000,
          "Specification": "MPEG4",
          "Bitrate": 64000
        }
      },
      "LanguageCodeControl": "FOLLOW_INPUT",
      "AudioSourceName": "Audio Selector 1",
      "StreamName": "English",
      "LanguageCode": "ENG"
    }
  ],
  "OutputSettings": {
    "HlsSettings": {
      "AudioGroupId": "program_audio",
      "AudioRenditionSets": "program_audio",
      "IFrameOnlyManifest": "EXCLUDE",
      "AudioTrackType": "ALTERNATE_AUDIO_AUTO_SELECT_DEFAULT"
    }
  },
  "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
      ]
    }
  }
}
```
Complex Example

},
"NameModifier": "_a1"
},

"AudioDescriptions": [
{
  "AudioTypeControl": "FOLLOW_INPUT",
  "CodecSettings": {
    "Codec": "AAC",
    "AacSettings": {
      "AudioDescriptionBroadcasterMix": "NORMAL",
      "RateControlMode": "CBR",
      "CodecProfile": "LC",
      "CodingMode": "CODING_MODE_2_0",
      "RawFormat": "NONE",
      "SampleRate": 48000,
      "Specification": "MPEG4",
      "Bitrate": 64000
    }
  },
  "LanguageCodeControl": "FOLLOW_INPUT",
  "AudioSourceName": "Audio Selector 2",
  "StreamName": "Spanish",
  "LanguageCode": "SPA"
},

"OutputSettings": {
  "HlsSettings": {
    "AudioGroupId": "program_audio",
    "AudioRenditionSets": "program_audio",
    "IFrameOnlyManifest": "EXCLUDE",
    "AudioTrackType": "ALTERNATE_AUDIO_AUTO_SELECT"
  }
},

"ContainerSettings": {
  "Container": "M3U8",
  "M3u8Settings": {
    "AudioFramesPerPes": 4,
    "PcrControl": "PCR_EVERY_PES_PACKET",
    "PmtPid": 480,
    "PrivateMetadataPid": 503,
    "ProgramNumber": 1,
    "PmtInterval": 0,
    "PmtInterval": 0,
    "Scte35Source": "NONE",
    "NielsenId3": "NONE",
    "TimedMetadata": "NONE",
    "VideoPid": 481,
    "AudioPids": [
      482,
      483,
      484,
      485,
      486,
      487,
      488,
      489,
      490,
      491,
      492
    ]
  }
},

"NameModifier": "_a2"
]
{ "AdAvailOffset": 0,
"Inputs": [
    {
        "AudioSelectors": {
            "Audio Selector 1": {
                "Offset": 0,
                "DefaultSelection": "DEFAULT",
                "ProgramSelection": 1,
                "SelectorType": "TRACK",
                "Tracks": [1]
            },
            "Audio Selector 2": {
                "Offset": 0,
                "DefaultSelection": "NOT_DEFAULT",
                "ProgramSelection": 1,
                "SelectorType": "TRACK",
                "Tracks": [2]
            }
        },
        "VideoSelector": {
            "ColorSpace": "FOLLOW"
        },
        "FilterEnable": "AUTO",
        "PsiControl": "USE_PSI",
        "FilterStrength": 0,
        "DeblockFilter": "DISABLED",
        "DenoiseFilter": "DISABLED",
        "TimecodeSource": "EMBEDDED",
        "FileInput": "s3://testbucket-input/720/test_file.mp4"
    }
],
"TimecodeConfig": {
    "Source": "ZEROBASED"
}
Important Notes

Versioned Endpoints

When you send requests programmatically, you must specify the API version you send the requests to. The current API version is 2017-08-29.

Therefore, you must construct your endpoint like this:

https://<account-specific-digits>.mediaconvert.<region>.amazonaws.com/2017-08-29/<resource>/<parameter>

For example:

https://abcd1234.mediaconvert.region-name-1.amazonaws.com/2017-08-29/queues/my-queue

Using the AWS CLI

When you send requests using the AWS CLI, use PascalCase for all properties. For example, if you used the properties `settings` and `timecodeConfig` in your API call, you must change those to `Settings` and `TimecodeConfig` for your CLI call. This is required because the CLI is built on Python, which uses PascalCase for properties.

About the Resources Section of This Guide

The Resources section of this guide provides two topics for each high level resource. The first topic describes the resource and the second provides information for working on a specific instance of the resource. For example, jobs has a topic, Jobs, and a topic, A Specific Job. Within each topic you'll find the available REST methods for the resource and the schema for your request body.

Note: The schemas provided in this guide are not working examples. For example job settings, see the AWS Elemental MediaConvert User Guide example job settings.
Resources

The AWS Elemental MediaConvert REST API includes the following resources.

Topics
- Endpoints (p. 44)
- JobTemplates (p. 46)
- JobTemplates name (p. 198)
- Jobs (p. 351)
- Jobs id (p. 507)
- Presets (p. 640)
- Presets name (p. 754)
- Queues (p. 868)
- Queues name (p. 874)

Endpoints

URI

/2017-08-29/endpoints

HTTP Methods

POST

Operation ID: DescribeEndpoints

Send an request with an empty body to the regional API endpoint to get your account API endpoint.

Responses

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>DescribeEndpointsResponse</td>
<td>200 response</td>
</tr>
<tr>
<td>400</td>
<td>ExceptionBody (p. 45)</td>
<td>BadRequestException</td>
</tr>
<tr>
<td>500</td>
<td>ExceptionBody (p. 45)</td>
<td>InternalServiceException</td>
</tr>
<tr>
<td>403</td>
<td>ExceptionBody (p. 45)</td>
<td>AccessDeniedException</td>
</tr>
<tr>
<td>404</td>
<td>ExceptionBody (p. 45)</td>
<td>ResourceNotFoundException</td>
</tr>
<tr>
<td>429</td>
<td>ExceptionBody (p. 45)</td>
<td>LimitExceededException</td>
</tr>
<tr>
<td>409</td>
<td>ExceptionBody (p. 45)</td>
<td>ResourceInUseException</td>
</tr>
</tbody>
</table>
Schemas

Request Bodies

Example POST

```json
{
  "nextToken (p. 45)": "string",
  "maxResults (p. 45)": integer
}
```

Response Bodies

Example DescribeEndpointsResponse

```json
{
  "endpoints (p. 45)": [
    {
      "url (p. 46)": "string"
    },
    "nextToken (p. 46)": "string"
  ]
}
```

Example ExceptionBody

```json
{
  "message (p. 46)": "string"
}
```

Properties

DescribeEndpointsRequest

nextToken

Use this string, provided with the response to a previous request, to request the next batch of endpoints.

- **Type:** string
- **Required:** False

maxResults

Optional. Max number of endpoints, up to twenty, that will be returned at one time.

- **Type:** integer
- **Required:** False
- **Format:** int32

DescribeEndpointsResponse

endpoints

List of endpoints
Type: Array of type Endpoint (p. 46)
Required: False

nextToken
Use this string to request the next batch of endpoints.

Type: string
Required: False

Endpoint

url
URL of endpoint

Type: string
Required: False

ExceptionBody

message

Type: string
Required: False

JobTemplates

URI

/2017-08-29/jobTemplates

HTTP Methods

GET

Operation ID: ListJobTemplates

Retrieve a JSON array of up to twenty of your job templates. This will return the templates themselves, not just a list of them. To retrieve the next twenty templates, use the nextToken string returned with the array

Query Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>listBy</td>
<td>String</td>
<td>False</td>
<td></td>
</tr>
<tr>
<td>nextToken</td>
<td>String</td>
<td>False</td>
<td></td>
</tr>
<tr>
<td>maxResults</td>
<td>String</td>
<td>False</td>
<td></td>
</tr>
</tbody>
</table>
## Schemas

### Request Bodies

**Example GET**

```json
{
  "nextToken (p. 157)" : "string",
  "maxResults (p. 158)" : integer,
```
Example POST

```json
{
  "settings (p. 102)": {
    "timecodeConfig (p. 153)": {
      "timestampOffset (p. 191)": "string",
      "anchor (p. 191)": "string",
      "start (p. 192)": "string",
      "source (p. 192)": enum
    },
    "adAvailOffset (p. 153)": integer,
    "outputGroups (p. 153)": [
      {
        "outputs (p. 182)": [
          {
            "extension (p. 181)": "string",
            "videoDescription (p. 181)": {
              "fixedAfd (p. 194)": integer,
              "scalingBehavior (p. 194)": enum,
              "respondToDoAfd (p. 194)": enum,
              "codecSettings (p. 194)": {
                "h265Settings (p. 193)": {
                  "slices (p. 130)": integer,
                  "minIInterval (p. 130)": integer,
                  "parNumerator (p. 130)": integer,
                  "flickerAdaptiveQuantization (p. 131)": enum,
                  "gopSizeUnits (p. 131)": enum,
                  "hrdBufferSize (p. 131)": integer,
                  "qualityTuningLevel (p. 131)": enum,
                  "maxBitrate (p. 131)": integer,
                  "bitrate (p. 131)": integer,
                  "spatialAdaptiveQuantization (p. 131)": enum,
                  "sampleAdaptiveOffsetFilterMode (p. 131)": enum,
                  "temporalAdlids (p. 132)": enum,
                  "slowPal (p. 132)": enum,
                  "writeMp4PackagingType (p. 132)": enum,
                  "codecProfile (p. 132)": enum,
                  "alternateTransferFunctionSei (p. 132)": enum,
                  "unregisteredSeiTimecode (p. 132)": enum,
                  "framerateControl (p. 132)": enum,
                  "televine (p. 132)": enum,
                  "codecLevel (p. 132)": enum,
                  "framerateConversionAlgorithm (p. 133)": enum,
                  "numberReferenceFrames (p. 133)": integer,
                  "temporalAdaptiveQuantization (p. 133)": enum,
                  "hrdBufferInitialFillPercentage (p. 133)": integer,
                  "framerateNumerator (p. 133)": integer,
                  "framerateDenominator (p. 133)": integer,
                  "adaptiveQuantization (p. 134)": enum,
                  "interlaceMode (p. 134)": enum,
                  "gopSize (p. 134)": number,
                  "gopBReference (p. 134)": enum,
                  "parDenominator (p. 134)": integer,
                  "sceneChangeDetect (p. 134)": enum,
                  "parControl (p. 135)": enum,
                  "rateControlMode (p. 135)": enum
                }
              }
            }
          }
        ]
      }
    ]
  }
}
```
"codec (p. 193)": enum,
"proresSettings (p. 193)": {
  "framerateDenominator (p. 185)": integer,
  "slowPal (p. 185)": enum,
  "framerateControl (p. 185)": enum,
  "telecine (p. 185)": enum,
  "framerateConversionAlgorithm (p. 185)": enum,
  "interlaceMode (p. 186)": enum,
  "parNumerator (p. 186)": integer,
  "codecProfile (p. 186)": enum,
  "parDenominator (p. 186)": integer,
  "parControl (p. 186)": enum,
  "framerateNumerator (p. 186)": integer
},
"mpeg2Settings (p. 193)": {
  "minIInterval (p. 173)": integer,
  "parNumerator (p. 173)": integer,
  "gopSizeUnits (p. 173)": enum,
  "hrdBufferSize (p. 174)": integer,
  "qualityTuningLevel (p. 174)": enum,
  "maxBitrate (p. 174)": integer,
  "bitrate (p. 174)": integer,
  "spatialAdaptiveQuantization (p. 174)": enum,
  "slowPal (p. 174)": enum,
  "intraDcPrecision (p. 174)": enum,
  "codecProfile (p. 174)": enum,
  "softness (p. 175)": integer,
  "framerateControl (p. 175)": enum,
  "telecine (p. 175)": enum,
  "codecLevel (p. 175)": enum,
  "framerateConversionAlgorithm (p. 175)": enum,
  "temporalAdaptiveQuantization (p. 175)": enum,
  "hrdBufferInitialFillPercentage (p. 175)": integer,
  "framerateNumerator (p. 175)": integer,
  "gopClosedCadence (p. 176)": integer,
  "numberBFramesBetweenReferenceFrames (p. 176)": integer,
  "framerateDenominator (p. 176)": integer,
  "adaptiveQuantization (p. 176)": enum,
  "interlaceMode (p. 176)": enum,
  "gopSize (p. 176)": number,
  "syntax (p. 176)": enum,
  "parDenominator (p. 177)": integer,
  "sceneChangeDetect (p. 177)": enum,
  "parControl (p. 177)": enum,
  "rateControlMode (p. 177)": enum
},
"frameCaptureSettings (p. 194)": {
  "framerateDenominator (p. 118)": integer,
  "maxCaptures (p. 118)": integer,
  "framerateNumerator (p. 118)": integer,
  "quality (p. 118)": integer
},
"h264Settings (p. 194)": {
  "slices (p. 121)": integer,
  "minIInterval (p. 122)": integer,
  "parNumerator (p. 122)": integer,
  "flickerAdaptiveQuantization (p. 122)": enum,
  "gopSizeUnits (p. 122)": enum,
  "hrdBufferSize (p. 122)": integer,
  "qualityTuningLevel (p. 122)": enum,
  "maxBitrate (p. 122)": integer,
  "bitrate (p. 123)": integer,
  "spatialAdaptiveQuantization (p. 123)": enum,
  "slowPal (p. 123)": enum,
  "codecProfile (p. 123)": enum,
"unregisteredSeiTimecode (p. 123)": enum,
"softness (p. 123)": integer,
"framerateControl (p. 123)": enum,
"telecine (p. 123)": enum,
"codecLevel (p. 124)": enum,
"framerateConversionAlgorithm (p. 124)": enum,
"numberReferenceFrames (p. 124)": integer,
"temporalAdaptiveQuantization (p. 124)": enum,
"repeatPps (p. 124)": enum,
"hrdBufferInitialFillPercentage (p. 124)": integer,
"framerateNumerator (p. 124)": integer,
"gopClosedCadence (p. 124)": integer,
"numberBFramesBetweenReferenceFrames (p. 125)": integer,
"framerateDenominator (p. 125)": integer,
"entropyEncoding (p. 125)": enum,
"fieldEncoding (p. 125)": enum,
"adaptiveQuantization (p. 125)": enum,
"interlaceMode (p. 125)": enum,
"gopSize (p. 125)": number,
"gopBReference (p. 126)": enum,
"syntax (p. 126)": enum,
"parDenominator (p. 126)": integer,
"sceneChangeDetect (p. 126)": enum,
"parControl (p. 126)": enum,
"rateControlMode (p. 126)": enum
},
"afdSignaling (p. 194)": enum,
"colorMetadata (p. 194)": enum,
"timecodeInsertion (p. 194)": enum,
"width (p. 195)": integer,
"videoPreprocessors (p. 195)": {
  "noiseReducer (p. 196)": {
    "filter (p. 179)": enum,
    "filterSettings (p. 179)": {
      "strength (p. 180)": integer
    },
    "spatialFilterSettings (p. 179)": {
      "strength (p. 180)": integer,
      "postFilterSharpenStrength (p. 180)": integer,
      "speed (p. 180)": integer
    }
  },
  "timecodeBurnin (p. 196)": {
    "prefix (p. 190)": "string",
    "fontSize (p. 191)": integer,
    "position (p. 191)": enum
  },
  "colorCorrector (p. 196)": {
    "saturation (p. 99)": integer,
    "brightness (p. 99)": integer,
    "hdr10Metadata (p. 99)": {
      "redPrimaryY (p. 136)": integer,
      "greenPrimaryY (p. 136)": integer,
      "whitePointX (p. 136)": integer,
      "maxLuminance (p. 137)": integer,
      "greenPrimaryX (p. 137)": integer,
      "whitePointY (p. 137)": integer,
      "redPrimaryX (p. 137)": integer,
      "bluePrimaryY (p. 137)": integer,
      "bluePrimaryX (p. 138)": integer,
      "maxFrameAverageLightLevel (p. 138)": integer,
      "maxContentLightLevel (p. 138)": integer,
      "minLuminance (p. 138)": integer
    },
    "contrast (p. 100)": integer,
  },
"hue (p. 100)": integer,
"colorSpaceConversion (p. 100)": enum
},
"deinterlacer (p. 196)": {
"mode (p. 105)": enum,
"control (p. 105)": enum,
"algorithm (p. 105)": enum
},
"imageInserter (p. 196)": {
"insertableImages (p. 146)": [
{
"duration (p. 149)": integer,
"fadeOut (p. 150)": integer,
"imageY (p. 150)": integer,
"fadeIn (p. 150)": integer,
"imageX (p. 150)": integer,
"width (p. 150)": integer,
"startTime (p. 151)": "string",
"opacity (p. 151)": integer,
"layer (p. 151)": integer,
"height (p. 151)": integer,
"imageInserterInput (p. 151)": "string"
}
]
},
"antiAlias (p. 195)": enum,
"position (p. 195)": {
"width (p. 187)": integer,
"x (p. 187)": integer,
"y (p. 187)": integer,
"height (p. 187)": integer
},
"sharpness (p. 195)": integer,
"crop (p. 195)": {
"width (p. 187)": integer,
"x (p. 187)": integer,
"y (p. 187)": integer,
"height (p. 187)": integer
},
"dropFrameTimecode (p. 196)": enum,
"height (p. 196)": integer
},
"audioDescriptions (p. 181)": [
{
"audioTypeControl (p. 85)": enum,
"languageCodeControl (p. 85)": enum,
"remixSettings (p. 85)": {
"channelsOut (p. 187)": integer,
"channelMapping (p. 188)": {
"outputChannels (p. 95)": [
{
"inputChannels (p. 182)": [ integer
]
}
],
"channelsIn (p. 188)": integer
},
"audioType (p. 85)": integer,
"audioSourceName (p. 86)": "string",
"codecSettings (p. 86)": {
"codec (p. 84)": enum,
"wavSettings (p. 84)": {
"channels (p. 198)": integer,
"bitDepth (p. 198)": integer,
"format (p. 198)": enum,
"sampleRate (p. 198)": integer
},
"aacSettings (p. 84)": {
"vbrQuality (p. 80)": enum,
"codecProfile (p. 80)": enum,
"codingMode (p. 80)": enum,
"specification (p. 80)": enum,
"bitrate (p. 80)": integer,
"rawFormat (p. 80)": enum,
"rateControlMode (p. 80)": enum,
"sampleRate (p. 80)": integer,
"audioDescriptionBroadcasterMix (p. 81)": enum
},
"ac3Settings (p. 85)": {
"dynamicRangeCompressionProfile (p. 82)": enum,
"codingMode (p. 82)": enum,
"metadataControl (p. 82)": enum,
"bitrate (p. 82)": integer,
"lfeFilter (p. 83)": enum,
"bitstreamMode (p. 83)": enum,
"sampleRate (p. 83)": integer
},
"aiffSettings (p. 85)": {
"channels (p. 83)": integer,
"bitDepth (p. 83)": integer,
"sampleRate (p. 83)": integer
},
"eac3Settings (p. 85)": {
"dialnorm (p. 113)": integer,
"passthroughControl (p. 113)": enum,
"metadataControl (p. 113)": enum,
"bitrate (p. 113)": integer,
"dynamicRangeCompressionRf (p. 113)": enum,
"sampleRate (p. 113)": integer,
"surroundExMode (p. 113)": enum,
"ltRtSurroundMixLevel (p. 114)": number,
"dynamicRangeCompressionLine (p. 114)": enum,
"lfeControl (p. 114)": enum,
"codingMode (p. 114)": enum,
"surroundMode (p. 114)": enum,
"attenuationControl (p. 114)": enum,
"lfeFilter (p. 114)": enum,
"phaseControl (p. 114)": enum,
"ltRtCenterMixLevel (p. 114)": number,
"dcFilter (p. 115)": enum,
"bitstreamMode (p. 115)": enum,
"stereoDownmix (p. 115)": enum,
"loRoSurrondMixLevel (p. 115)": number,
"loRoCenterMixLevel (p. 115)": number
},
"mp2Settings (p. 85)": {
"channels (p. 169)": integer,
"bitrate (p. 170)": integer,
"sampleRate (p. 170)": integer
},
"languageCode (p. 86)": enum,
"streamName (p. 86)": "string",
"audioNormalizationSettings (p. 86)": {
"targetLkfs (p. 87)": number,
"algorithmControl (p. 88)": enum,
"loudnessLogging (p. 88)": enum,
"peakCalculation (p. 88)": enum,
"correctionGateLevel (p. 88)": integer,
"algorithm (p. 88)": enum
},
"customLanguageCode (p. 86)": "string"
]

"containerSettings (p. 181)": {
"container (p. 101)": enum,
"mp4Settings (p. 101)": {
"mp4MajorBrand (p. 170)": "string",
"moovPlacement (p. 171)": enum,
"cslgAtom (p. 171)": enum,
"freeSpaceBox (p. 171)": enum
},
"m3u8Settings (p. 101)": {
"pmtPid (p. 166)": integer,
"nielsenId3 (p. 166)": enum,
"pcrPid (p. 166)": integer,
"audioPids (p. 166)": [ integer
],
"audioFramesPerPES (p. 166)": integer,
"scte35Pid (p. 166)": integer,
"transportStreamId (p. 167)": integer,
"pcrControl (p. 167)": enum,
"videoPid (p. 167)": integer,
"privateMetadataPid (p. 167)": integer,
"pmtInterval (p. 167)": integer,
"patInterval (p. 167)": integer,
"programNumber (p. 168)": integer,
"timedMetadataPid (p. 168)": integer,
"timedMetadata (p. 168)": enum,
"scte35Source (p. 168)": enum
},
"m2tsSettings (p. 101)": {
"dvbTeletextPid (p. 160)": integer,
"nielsenId3 (p. 161)": enum,
"bitrate (p. 161)": integer,
"segmentationTime (p. 161)": number,
"audioPids (p. 161)": [ integer
],
"rateMode (p. 161)": enum,
"maxPcrInterval (p. 161)": integer,
"audioFramesPerPES (p. 161)": integer,
"ebpAudioInterval (p. 162)": enum,
"fragmentTime (p. 162)": number,
"scte35Pid (p. 162)": integer,
"pmtInterval (p. 162)": integer,
"privateMetadataPid (p. 162)": integer,
"segmentationStyle (p. 162)": enum,
"audioBufferModel (p. 162)": enum,
"programNumber (p. 163)": integer,
"dvbNitSettings (p. 163)": {
"networkName (p. 105)": "string",
"networkId (p. 106)": integer,
"nitInterval (p. 106)": integer
},
"timedMetadataPid (p. 163)": integer,
"scte35Source (p. 163)": enum,
"pmtPid (p. 163)": integer,
"bufferModel (p. 163)": enum,
"ebpPlacement (p. 163)": enum,
"dvbSdtSettings (p. 163)": {
"sdtInterval (p. 106)": integer,
"serviceName (p. 106)": "string"}
"serviceProviderName (p. 106)" : "string",
"outputSdt (p. 106)" : enum
},
"nullPacketBitrate (p. 164)" : number,
"pcrPid (p. 164)" : integer,
"minEbpInterval (p. 164)" : integer,
"transportStreamId (p. 164)" : integer,
"videoPid (p. 164)" : integer,
"pcrControl (p. 165)" : enum,
"esRateInPes (p. 165)" : enum,
"segmentationMarkers (p. 165)" : enum,
"dvbTdtSettings (p. 165)" : {
  "tdtInterval (p. 111)" : integer
},
"patInterval (p. 165)" : integer,
"dvbSubPids (p. 165)" : [
  integer
],
"movSettings (p. 101)" : {
  "reference (p. 169)" : enum,
  "paddingControl (p. 169)" : enum,
  "mpeg2FourCCControl (p. 169)" : enum,
  "cs1gAtom (p. 169)" : enum,
  "clapAtom (p. 169)" : enum
},
"f4vSettings (p. 101)" : {
  "moovPlacement (p. 117)" : enum
},
"preset (p. 181)" : "string",
"outputSettings (p. 182)" : {
  "hlsSettings (p. 184)" : {
    "iFrameOnlyManifest (p. 145)" : enum,
    "audioRenditionSets (p. 145)" : "string",
    "audioTrackType (p. 145)" : enum,
    "segmentModifier (p. 145)" : "string",
    "audioGroupId (p. 145)" : "string"
  }
},
"captionDescriptions (p. 182)" : [
  {
    "captionSelectorName (p. 92)" : "string",
    "languageDescription (p. 93)" : "string",
    "destinationSettings (p. 93)" : {
      "burninDestinationSettings (p. 93)" : {
        "xPosition (p. 89)" : integer,
        "backgroundColor (p. 89)" : enum,
        "teletextSpacing (p. 89)" : enum,
        "yPosition (p. 89)" : integer,
        "backgroundColor (p. 89)" : integer,
        "fontOpacity (p. 90)" : integer,
        "fontResolution (p. 90)" : integer,
        "shadowOpacity (p. 90)" : integer,
        "shadowYOffset (p. 90)" : integer,
        "outlineSize (p. 91)" : integer,
        "outlineColor (p. 91)" : enum,
        "shadowXOffset (p. 91)" : integer,
        "alignment (p. 91)" : enum,
        "shadowColor (p. 91)" : enum,
        "fontColor (p. 91)" : enum
      },
      "teletextDestinationSettings (p. 93)" : {
        "pageNumber (p. 190)" : "string"
      }
    }
  }
}
"ttmlDestinationSettings (p. 93)": {
  "stylePassthrough (p. 193)": enum
},
"destinationType (p. 93)": enum,
"dvbSubDestinationSettings (p. 94)": {
  "xPosition (p. 107)": integer,
  "backgroundColor (p. 107)": enum,
  "teletextSpacing (p. 107)": enum,
  "yPosition (p. 107)": integer,
  "backgroundColorOpacity (p. 107)": integer,
  "fontOpacity (p. 107)": integer,
  "fontResolution (p. 108)": integer,
  "shadowOpacity (p. 108)": integer,
  "shadowYOffset (p. 108)": integer,
  "outlineSize (p. 108)": integer,
  "outlineColor (p. 108)": enum,
  "fontSize (p. 108)": integer,
  "shadowXOffset (p. 109)": integer,
  "alignment (p. 109)": enum,
  "shadowColor (p. 109)": enum,
  "fontColor (p. 109)": enum
},
"sccDestinationSettings (p. 94)": {
  "framerate (p. 189)": enum
},
"languageCode (p. 93)": enum,
"customLanguageCode (p. 93)": "string"
],
"nameModifier (p. 182)": "string"}
"audioDeduplication (p. 178)": enum,
"manifestEncoding (p. 178)": enum,
"destination (p. 179)": "string"
},
"cmafGroupSettings (p. 183)": {
  "writeHlsManifest (p. 96)": enum,
  "writeDashManifest (p. 96)": enum,
  "fragmentLength (p. 96)": integer,
  "streamInfResolution (p. 97)": enum,
  "minBufferTime (p. 97)": integer,
  "clientCache (p. 97)": enum,
  "codecSpecification (p. 97)": enum,
  "destination (p. 97)": "string",
  "segmentControl (p. 97)": enum,
  "baseUrl (p. 97)": "string",
  "encryption (p. 97)": {
    "initializationVectorInManifest (p. 95)": enum,
    "constantInitializationVector (p. 96)": "string",
    "staticKeyProvider (p. 96)": {
      "keyFormatVersions (p. 189)": "string",
      "keyFormat (p. 189)": "string",
      "staticKeyValue (p. 190)": "string",
      "url (p. 190)": "string"
    },
    "type (p. 96)": enum,
  },
  "manifestCompression (p. 98)": enum,
  "segmentLength (p. 98)": integer,
  "manifestDurationFormat (p. 98)": enum
},
"type (p. 183)": enum,
"hlsGroupSettings (p. 183)": {
  "segmentsPerSubdirectory (p. 141)": integer,
  "streamInfResolution (p. 141)": enum,
  "timestampDeltaMilliseconds (p. 141)": integer,
  "outputSelection (p. 141)": enum,
  "captionLanguageMappings (p. 141)": [{
    "languageDescription (p. 139)": "string",
    "captionChannel (p. 139)": integer,
    "languageCode (p. 139)": enum,
    "customLanguageCode (p. 139)": "string"
  }]
},
"clientCache (p. 142)": enum,
"codecSpecification (p. 142)": enum,
"destination (p. 142)": "string",
"timedMetadataId3Frame (p. 142)": enum,
"segmentControl (p. 142)": enum,
"timedMetadataId3Period (p. 142)": integer,
"captionLanguageSetting (p. 142)": enum,
"minSegmentLength (p. 142)": integer,
"directoryStructure (p. 143)": enum,
"programDateTime (p. 143)": enum,
"baseUrl (p. 143)": "string",
"encryption (p. 143)": {
  "initializationVectorInManifest (p. 140)": enum,
  "constantInitializationVector (p. 140)": "string",
  "staticKeyProvider (p. 140)": {
    "keyFormatVersions (p. 189)": "string",
    "keyFormat (p. 189)": "string",
    "staticKeyValue (p. 190)": "string",
    "url (p. 190)": "string"
  },
  "type (p. 140)": enum,
"encryptionMethod (p. 140)": enum,
"spekeKeyProvider (p. 141)": {
  "resourceId (p. 189)": "string",
  "systemIds (p. 189)": [
    "string"
  ],
  "url (p. 189)": "string"
},
"sdMarkers (p. 143)": [enum
],
"programDateTimePeriod (p. 143)": integer,
"manifestCompression (p. 143)": enum,
"segmentLength (p. 143)": integer,
"manifestDurationFormat (p. 144)": enum
},
"name (p. 182)": "string",
"customName (p. 183)": "string"
},
"nielsenConfiguration (p. 153)": {
  "distributorId (p. 179)": "string",
  "breakoutCode (p. 179)": integer
},
"inputs (p. 154)": [
  {
    "audioSelectors (p. 148)": {
    },
    "audioSelectorGroups (p. 148)": {
    },
    "filterEnable (p. 148)": enum,
    "deblockFilter (p. 148)": enum,
    "videoSelector (p. 148)": {
      "colorSpace (p. 197)": enum,
      "hdr10Metadata (p. 197)": {
        "redPrimaryY (p. 136)": integer,
        "greenPrimaryY (p. 136)": integer,
        "whitePointX (p. 136)": integer,
        "maxLuminance (p. 137)": integer,
        "greenPrimaryX (p. 137)": integer,
        "whitePointY (p. 137)": integer,
        "redPrimaryX (p. 137)": integer,
        "bluePrimaryX (p. 137)": integer,
        "bluePrimaryY (p. 138)": integer,
        "maxFrameAverageLightLevel (p. 138)": integer,
        "maxContentLightLevel (p. 138)": integer,
        "minLuminance (p. 138)": integer
      },
      "programNumber (p. 197)": integer,
      "pid (p. 197)": integer,
      "colorSpaceUsage (p. 197)": enum
    },
    "filterStrength (p. 148)": integer,
    "programNumber (p. 148)": integer,
    "timecodeSource (p. 149)": enum,
    "denoiseFilter (p. 149)": enum,
    "captionSelectors (p. 149)": {
    },
    "inputClippings (p. 149)": [
      {
        "startTimecode (p. 146)": "string",
        "endTimecode (p. 147)": "string"
      }
    ]
  }
]}
"psiControl (p. 149)": enum
},
"timedMetadataInsertion (p. 154)": {
  "id3Insertions (p. 192)": [
    {
      "id3 (p. 146)": "string",
      "timecode (p. 146)": "string"
    }
  ]
},
"availBlanking (p. 154)": {
  "availBlankingImage (p. 88)": "string"
},
"name (p. 102)": "string",
"description (p. 102)": "string",
"category (p. 102)": "string",
"queue (p. 102)": "string"
}

Response Bodies

Example ListJobTemplatesResponse

```json
{
  "nextToken (p. 158)": "string",
  "jobTemplates (p. 158)": [
    {
      "createdAt (p. 151)": "string",
      "lastUpdated (p. 152)": "string",
      "settings (p. 152)": {
        "timecodeConfig (p. 153)": {
          "timestampOffset (p. 191)": "string",
          "anchor (p. 191)": "string",
          "start (p. 192)": "string",
          "source (p. 192)": enum
        },
        "adAvailOffset (p. 153)": integer,
        "outputGroups (p. 153)": [
          {
            "outputs (p. 182)": [
              {
                "extension (p. 181)": "string",
                "videoDescription (p. 181)": {
                  "fixedAfd (p. 194)": integer,
                  "scalingBehavior (p. 194)": enum,
                  "respondToAfd (p. 194)": enum,
                  "codecSettings (p. 194)": {
                    "h265Settings (p. 194)": {
                      "slices (p. 130)": integer,
                      "minInterval (p. 130)": integer,
                      "parNumerator (p. 130)": integer,
                      "flickerAdaptiveQuantization (p. 131)": enum,
                      "gopSizeUnits (p. 131)": enum,
                      "hrdBufferSize (p. 131)": integer,
                      "qualityTuningLevel (p. 131)": enum,
                      "maxBitrate (p. 131)": integer,
                      "bitrate (p. 131)": integer,
                      "spatialAdaptiveQuantization (p. 131)": enum,
                      "sampleAdaptiveOffsetFilterMode (p. 131)": enum,
                      "temporalIds (p. 132)": enum,
                      "slowPal (p. 132)": enum,
                      "tiles (p. 132)": enum,
```
"writeMp4PackagingType (p. 132)": enum,
"codecProfile (p. 132)": enum,
"alternateTransferFunctionSei (p. 132)": enum,
"unregisteredSeiTimecode (p. 132)": enum,
"framerateControl (p. 132)": enum,
"telecine (p. 132)": enum,
"codecLevel (p. 132)": enum,
"framerateConversionAlgorithm (p. 133)": enum,
"numberReferenceFrames (p. 133)": integer,
"temporalAdaptiveQuantization (p. 133)": enum,
"hrdBufferInitialFillPercentage (p. 133)": integer,
"framerateNumerator (p. 133)": integer,
"gopClosedCadence (p. 133)": integer,
"numberBFramesBetweenReferenceFrames (p. 133)": integer,
"framerateDenominator (p. 134)": integer,
"adaptiveQuantization (p. 134)": enum,
"interlaceMode (p. 134)": enum,
"gopSize (p. 134)": integer,
"gopReference (p. 134)": integer,
"parDenominator (p. 134)": integer,
"sceneChangeDetect (p. 134)": enum,
"parControl (p. 135)": enum,
"rateControlMode (p. 135)": enum,
"codec (p. 193)": enum,
"proresSettings (p. 193)": {
"framerateDenominator (p. 185)": integer,
"slowPal (p. 185)": enum,
"framerateControl (p. 185)": enum,
"telecine (p. 185)": enum,
"framerateConversionAlgorithm (p. 185)": enum,
"interlaceMode (p. 186)": enum,
"parNumerator (p. 186)": integer,
"codecProfile (p. 186)": enum,
"parDenominator (p. 186)": integer,
"parControl (p. 186)": enum,
"framerateNumerator (p. 186)": integer
},
"mpeg2Settings (p. 193)": {
"minIInterval (p. 173)": integer,
"parNumerator (p. 173)": integer,
"gopSizeUnits (p. 173)": enum,
"hrdBufferSize (p. 174)": integer,
"qualityTuningLevel (p. 174)": enum,
"maxBitrate (p. 174)": integer,
"bitrate (p. 174)": integer,
"spatialAdaptiveQuantization (p. 174)": enum,
"slowPal (p. 174)": enum,
"intraDcPrecision (p. 174)": enum,
"codecProfile (p. 174)": enum,
"softness (p. 175)": integer,
"framerateControl (p. 175)": enum,
"telecine (p. 175)": enum,
"codecLevel (p. 175)": enum,
"framerateConversionAlgorithm (p. 175)": enum,
"temporalAdaptiveQuantization (p. 175)": enum,
"hrdBufferInitialFillPercentage (p. 175)": integer,
"framerateNumerator (p. 175)": integer,
"gopClosedCadence (p. 176)": integer,
"numberBFramesBetweenReferenceFrames (p. 176)": integer,
"framerateDenominator (p. 176)": integer,
"adaptiveQuantization (p. 176)": enum,
"interlaceMode (p. 176)": enum,
"gopSize (p. 176)": integer,
"syntax (p. 176)": enum,
"parDenominator (p. 177)": integer,
"sceneChangeDetect (p. 177)": enum,
"parControl (p. 177)": enum,
"rateControlMode (p. 177)": enum,
",
"frameCaptureSettings (p. 194)": {
  "framerateDenominator (p. 118)": integer,
  "maxCaptures (p. 118)": integer,
  "framerateNumerator (p. 118)": integer,
  "quality (p. 118)": integer,
",
"h264Settings (p. 194)": {
  "slices (p. 121)": integer,
  "minIInterval (p. 122)": integer,
  "parNumerator (p. 122)": integer,
  "flickerAdaptiveQuantization (p. 122)": enum,
  "gopSizeUnits (p. 122)": enum,
  "hrdBufferSize (p. 122)": integer,
  "qualityTuningLevel (p. 122)": enum,
  "maxBitrate (p. 122)": integer,
  "bitrate (p. 123)": integer,
  "spatialAdaptiveQuantization (p. 123)": enum,
  "slowPal (p. 123)": enum,
  "codecProfile (p. 123)": enum,
  "unregisteredSeiTimecode (p. 123)": enum,
  "softness (p. 123)": integer,
  "framerateControl (p. 123)": enum,
  "telecine (p. 123)": enum,
  "codecLevel (p. 124)": enum,
  "framerateConversionAlgorithm (p. 124)": enum,
  "numberReferenceFrames (p. 124)": integer,
  "temporalAdaptiveQuantization (p. 124)": enum,
  "repeatPps (p. 124)": enum,
  "hrdBufferInitialFillPercentage (p. 124)": integer,
  "framerateNumerator (p. 124)": integer,
  "gopClosedCadence (p. 124)": integer,
  "numberBFramesBetweenReferenceFrames (p. 125)": integer,
  "framerateDenominator (p. 125)": integer,
  "entropyEncoding (p. 125)": enum,
  "fieldEncoding (p. 125)": enum,
  "adaptiveQuantization (p. 125)": enum,
  "interlaceMode (p. 125)": enum,
  "gopSize (p. 125)": number,
  "gopBReference (p. 126)": enum,
  "syntax (p. 126)": enum,
  "parDenominator (p. 126)": integer,
  "sceneChangeDetect (p. 126)": enum,
  "parControl (p. 126)": integer,
  "rateControlMode (p. 126)": enum
}
",
"afdSignaling (p. 194)": enum,
"colorMetadata (p. 194)": enum,
"timecodeInsertion (p. 194)": enum,
"width (p. 195)": integer,
"videoPreprocessors (p. 195)": {
  "noiseReducer (p. 196)": {
    "filter (p. 179)": enum,
    "filterSettings (p. 179)": {
      "strength (p. 180)": integer
    }
  },
  "spatialFilterSettings (p. 179)": {
    "strength (p. 180)": integer,
    "postFilterSharpenStrength (p. 180)": integer,
    "speed (p. 180)": integer
  }
}
"timecodeBurnin (p. 196)": {
  "prefix (p. 190)": "string",
  "fontSize (p. 191)": integer,
  "position (p. 191)": enum
},
"colorCorrector (p. 196)": {
  "saturation (p. 99)": integer,
  "brightness (p. 99)": integer,
  "hdr10Metadata (p. 99)": {
    "redPrimaryY (p. 136)": integer,
    "greenPrimaryY (p. 136)": integer,
    "whitePointX (p. 136)": integer,
    "maxLuminance (p. 137)": integer,
    "greenPrimaryX (p. 137)": integer,
    "whitePointY (p. 137)": integer,
    "redPrimaryX (p. 137)": integer,
    "bluePrimaryX (p. 137)": integer,
    "bluePrimaryY (p. 138)": integer,
    "maxFrameAverageLightLevel (p. 138)": integer,
    "maxContentLightLevel (p. 138)": integer,
    "minLuminance (p. 138)": integer
  },
  "contrast (p. 100)": integer,
  "hue (p. 100)": integer,
  "colorSpaceConversion (p. 100)": enum
},
"deinterlacer (p. 196)": {
  "mode (p. 105)": enum,
  "control (p. 105)": enum,
  "algorithm (p. 105)": enum
},
"imageInserter (p. 196)": {
  "insertableImages (p. 146)": [
    {
      "duration (p. 149)": integer,
      "fadeOut (p. 150)": integer,
      "imageY (p. 150)": integer,
      "fadeIn (p. 150)": integer,
      "imageX (p. 150)": integer,
      "width (p. 150)": integer,
      "startTime (p. 151)": "string",
      "opacity (p. 151)": integer,
      "layer (p. 151)": integer,
      "height (p. 151)": integer,
      "imageInserterInput (p. 151)": "string"
    }
  ]
},
"antiAlias (p. 195)": enum,
"position (p. 195)": {
  "width (p. 187)": integer,
  "x (p. 187)": integer,
  "y (p. 187)": integer,
  "height (p. 187)": integer
},
"sharpness (p. 195)": integer,
"crop (p. 195)": {
  "width (p. 187)": integer,
  "x (p. 187)": integer,
  "y (p. 187)": integer,
  "height (p. 187)": integer
},
"dropFrameTimecode (p. 196)": enum,
"height (p. 196)": integer
}
"audioDescriptions (p. 181)": [
  { "audioTypeControl (p. 85)": enum,
    "languageCodeControl (p. 85)": enum,
    "remixSettings (p. 85)": {
      "channelsOut (p. 187)": integer,
      "channelMapping (p. 188)": {
        "outputChannels (p. 95)": [
          
        ]
      }
    }
  },
  "channelsIn (p. 188)": integer
],
"audioType (p. 85)": integer,
"audioSourceName (p. 86)": string,
"codecSettings (p. 86)": {
  "codec (p. 84)": enum,
  "wavSettings (p. 84)": {
    "channels (p. 198)": integer,
    "bitDepth (p. 198)": integer,
    "format (p. 198)": enum,
    "sampleRate (p. 198)": integer
  },
  "aacSettings (p. 84)": {
    "vbrQuality (p. 80)": enum,
    "codecProfile (p. 80)": enum,
    "codingMode (p. 80)": enum,
    "specification (p. 80)": enum,
    "bitrate (p. 80)": integer,
    "rawFormat (p. 80)": enum,
    "rateControlMode (p. 80)": enum,
    "sampleRate (p. 80)": integer,
    "audioDescriptionBroadcasterMix (p. 81)": enum
  },
  "ac3Settings (p. 85)": {
    "dynamicRangeCompressionProfile (p. 82)": enum,
    "dialnorm (p. 82)": integer,
    "codingMode (p. 82)": enum,
    "metadataControl (p. 82)": enum,
    "bitrate (p. 82)": integer,
    "lfeFilter (p. 83)": enum,
    "bitstreamMode (p. 83)": enum,
    "sampleRate (p. 83)": integer
  },
  "aiffSettings (p. 85)": {
    "channels (p. 83)": integer,
    "bitDepth (p. 83)": integer,
    "sampleRate (p. 83)": integer
  },
  "eac3Settings (p. 85)": {
    "dialnorm (p. 113)": integer,
    "passthroughControl (p. 113)": enum,
    "metadataControl (p. 113)": enum,
    "bitrate (p. 113)": integer,
    "dynamicRangeCompressionRf (p. 113)": enum,
    "sampleRate (p. 113)": integer,
    "surroundExMode (p. 113)": enum,
    "l1rtSurroundMixLevel (p. 114)": number,
    "dynamicRangeCompressionLine (p. 114)": enum,
    "lfeControl (p. 114)": enum,
    "codingMode (p. 114)": enum,
    "surroundMode (p. 114)": enum,
  }
],
"62"
"attenuationControl (p. 114)" : enum,
"lfeFilter (p. 114)" : enum,
"phaseControl (p. 114)" : enum,
"ltRtCenterMixLevel (p. 114)" : number,
"dcFilter (p. 115)" : enum,
"bitstreamMode (p. 115)" : enum,
"stereoDownmix (p. 115)" : enum,
"loRoSurroundMixLevel (p. 115)" : number,
"loRoCenterMixLevel (p. 115)" : number
},
"mp2Settings (p. 85)" : {
  "channels (p. 169)" : integer,
  "bitrate (p. 170)" : integer,
  "sampleRate (p. 170)" : integer
},
"languageCode (p. 86)" : enum,
"streamName (p. 86)" : "string",
"audioNormalizationSettings (p. 86)" : {
  "targetLkfs (p. 87)" : number,
  "algorithmControl (p. 88)" : enum,
  "loudnessLogging (p. 88)" : enum,
  "peakCalculation (p. 88)" : enum,
  "correctionGateLevel (p. 88)" : integer,
  "algorithm (p. 88)" : enum
},
"customLanguageCode (p. 86)" : "string"
},
"containerSettings (p. 181)" : {
  "container (p. 101)" : enum,
  "mp4Settings (p. 101)" : {
    "mp4MajorBrand (p. 170)" : "string",
    "moovPlacement (p. 171)" : enum,
    "csigAtom (p. 171)" : enum,
    "freeSpaceBox (p. 171)" : enum
  },
  "m3u8Settings (p. 101)" : {
    "pmtPid (p. 166)" : integer,
    "nielsenId3 (p. 166)" : enum,
    "pcrPid (p. 166)" : integer,
    "audioPids (p. 166)" : [ integer
  ],
  "audioFramesPerPes (p. 166)" : integer,
  "scte35Pid (p. 166)" : integer,
  "transportStreamId (p. 167)" : integer,
  "pcrControl (p. 167)" : enum,
  "videoPid (p. 167)" : integer,
  "privateMetadataPid (p. 167)" : integer,
  "pmtInterval (p. 167)" : integer,
  "patInterval (p. 167)" : integer,
  "programNumber (p. 168)" : integer,
  "timedMetadataPid (p. 168)" : integer,
  "timedMetadata (p. 168)" : enum,
  "scte35Source (p. 168)" : enum
},
"m2tsSettings (p. 101)" : {
  "dvbTeletextPid (p. 160)" : integer,
  "nielsenId3 (p. 161)" : enum,
  "bitrate (p. 161)" : integer,
  "segmentationTime (p. 161)" : number,
  "audioPids (p. 161)" : [ integer
  ],
  "rateMode (p. 161)" : enum,
"maxPcrInterval (p. 161)": integer,
"audioFramesPerPes (p. 161)": integer,
"ebpAudioInterval (p. 162)": enum,
"fragmentTime (p. 162)": number,
"scte35Pid (p. 162)": integer,
"pmtInterval (p. 162)": integer,
"privateMetadataPid (p. 162)": integer,
"segmentationStyle (p. 162)": enum,
"audioBufferModel (p. 162)": enum,
"programNumber (p. 163)": integer,
"dvbNitSettings (p. 163)": {
  "networkName (p. 105)": "string",
  "networkId (p. 106)": integer,
  "nitInterval (p. 106)": integer
},
"timedMetadataPid (p. 163)": integer,
"scte35Source (p. 163)": enum,
"pmtPid (p. 163)": integer,
"bufferModel (p. 163)": enum,
"ebpPlacement (p. 163)": enum,
"dvbSdtSettings (p. 163)": {
  "sdtInterval (p. 106)": integer,
  "serviceName (p. 106)": "string",
  "serviceProviderName (p. 106)": "string",
  "outputSdt (p. 106)": enum
},
"nullPacketBitrate (p. 164)": number,
"pcrPid (p. 164)": integer,
"minEbpInterval (p. 164)": integer,
"transportStreamId (p. 164)": integer,
"videoPid (p. 164)": integer,
"pcrControl (p. 165)": enum,
"esRateInPes (p. 165)": enum,
"segmentationMarkers (p. 165)": enum,
"dvbTdtSettings (p. 165)": {
  "tdtInterval (p. 111)": integer
},
"patInterval (p. 165)": integer,
"dvbSubPids (p. 165)": [integer
],
"movSettings (p. 101)": {
  "reference (p. 169)": enum,
  "paddingControl (p. 169)": enum,
  "mpeg2FourCCControl (p. 169)": enum,
  "cslgAtom (p. 169)": enum,
  "clapAtom (p. 169)": enum
},
"f4vSettings (p. 101)": {
  "moovPlacement (p. 117)": enum
},
"preset (p. 181)": "string",
"outputSettings (p. 182)": {
  "hlsSettings (p. 184)": {
    "iFrameOnlyManifest (p. 145)": enum,
    "audioRenditionSets (p. 145)": "string",
    "audioTrackType (p. 145)": enum,
    "segmentModifier (p. 145)": "string",
    "audioGroupId (p. 145)": "string"
  }
},
"captionDescriptions (p. 182)": [
  {
    "captionSelectorName (p. 92)": "string"
"languageDescription (p. 93)" : "string",
"destinationSettings (p. 93)" : {
  "burninDestinationSettings (p. 93)" : {
    "xPosition (p. 89)" : integer,
    "backgroundColor (p. 89)" : enum,
    "teletextSpacing (p. 89)" : enum,
    "yPosition (p. 89)" : integer,
    "backgroundColor (p. 89)" : enum,
    "teletextSpacing (p. 89)" : enum,
    "fontOpacity (p. 90)" : integer,
    "fontResolution (p. 90)" : integer,
    "shadowOpacity (p. 90)" : integer,
    "shadowYOffset (p. 90)" : integer,
    "outlineSize (p. 90)" : integer,
    "outlineColor (p. 90)" : enum,
    "fontSize (p. 90)" : integer,
    "shadowXOffset (p. 90)" : integer,
    "alignment (p. 90)" : enum,
    "shadowColor (p. 90)" : enum,
    "fontColor (p. 90)" : enum
  },
  "teletextDestinationSettings (p. 93)" : {
    "pageNumber (p. 190)" : "string"
  },
  "ttmlDestinationSettings (p. 93)" : {
    "stylePassthrough (p. 193)" : enum
  },
  "destinationType (p. 93)" : enum,
  "dvbSubDestinationSettings (p. 94)" : {
    "xPosition (p. 107)" : integer,
    "backgroundColor (p. 107)" : enum,
    "teletextSpacing (p. 107)" : enum,
    "yPosition (p. 107)" : integer,
    "backgroundColor (p. 107)" : enum,
    "teletextSpacing (p. 107)" : enum,
    "fontOpacity (p. 107)" : integer,
    "fontResolution (p. 107)" : integer,
    "shadowOpacity (p. 107)" : integer,
    "shadowXOffset (p. 107)" : integer,
    "alignment (p. 107)" : enum,
    "shadowColor (p. 107)" : enum,
    "fontColor (p. 107)" : enum
  },
  "sccDestinationSettings (p. 94)" : {
    "framerate (p. 189)" : enum
  },
  "languageCode (p. 93)" : enum,
  "customLanguageCode (p. 93)" : "string"
},

"nameModifier (p. 182)" : "string"
}
]
"outputGroupSettings (p. 182)" : {
  "dashIsoGroupSettings (p. 183)" : {
    "fragmentLength (p. 103)" : integer,
    "baseUrl (p. 103)" : "string",
    "minBufferTime (p. 103)" : integer,
    "encryption (p. 103)" : {
      "spekeKeyProvider (p. 103)" : {
        "resourceId (p. 189)" : "string",
        "systemIds (p. 189)" : [
          "string"
        ]
      },
      "encryptionKeyId (p. 189)" : "string"
    }
  },
  "httpGroupSettings (p. 183)" : {
    "baseUrl (p. 103)" : "string",
    "minBufferTime (p. 103)" : integer,
    "encryption (p. 103)" : {
      "spekeKeyProvider (p. 103)" : {
        "resourceId (p. 189)" : "string",
        "systemIds (p. 189)" : [
          "string"
        ]
      },
      "encryptionKeyId (p. 189)" : "string"
    },
    " mariage"
},
    "url (p. 189)": "string"
  },
  "destination (p. 103)": "string",
  "segmentLength (p. 104)": integer,
  "segmentControl (p. 104)": enum,
  "hbbtvCompliance (p. 104)": enum
},
  "fileGroupSettings (p. 183)": {
    "destination (p. 117)": "string"
  },
  "msSmoothGroupSettings (p. 183)": {
    "fragmentLength (p. 178)": integer,
    "encryption (p. 178)": {
      "spekeKeyProvider (p. 178)": {
        "resourceId (p. 189)": "string",
        "systemIds (p. 189)": [
          "string"
        ],
        "url (p. 189)": "string"
      }
    },
    "audioDeduplication (p. 178)": enum,
    "manifestEncoding (p. 178)": enum,
    "destination (p. 179)": "string"
  },
  "cmafGroupSettings (p. 183)": {
    "writeHlsManifest (p. 96)": enum,
    "writeDashManifest (p. 96)": enum,
    "fragmentLength (p. 96)": integer,
    "streamInfResolution (p. 97)": enum,
    "minBufferTime (p. 97)": integer,
    "clientCache (p. 97)": enum,
    "codecSpecification (p. 97)": enum,
    "destination (p. 97)": "string",
    "segmentControl (p. 97)": enum,
    "baseUrl (p. 97)": "string",
    "encryption (p. 97)": {
      "initializationVectorInManifest (p. 95)": enum,
      "constantInitializationVector (p. 96)": "string",
      "staticKeyProvider (p. 96)": {
        "keyFormatInitializationVector (p. 189)": "string",
        "keyFormat (p. 189)": "string",
        "staticKeyValue (p. 190)": "string",
        "url (p. 190)": "string"
      },
      "type (p. 96)": enum,
      "encryptionMethod (p. 96)": enum
    },
    "manifestCompression (p. 98)": enum,
    "segmentLength (p. 98)": integer,
    "manifestDurationFormat (p. 98)": enum
  },
  "type (p. 183)": enum,
  "hlsGroupSettings (p. 183)": {
    "segmentsPerSubdirectory (p. 141)": integer,
    "streamInfResolution (p. 141)": enum,
    "timestampDeltaMilliseconds (p. 141)": integer,
    "outputSelection (p. 141)": enum,
    "captionLanguageMappings (p. 141)": [
      {
        "languageDescription (p. 139)": "string",
        "captionChannel (p. 139)": integer,
        "languageCode (p. 139)": enum,
        "customLanguageCode (p. 139)": "string"}}}
"clientCache (p. 142)": enum,
"codecSpecification (p. 142)": enum,
"destination (p. 142)": "string",
"timedMetadataId3Frame (p. 142)": enum,
"segmentControl (p. 142)": enum,
"timedMetadataId3Period (p. 142)": integer,
"captionLanguageSetting (p. 142)": enum,
"minSegmentLength (p. 142)": integer,
"directoryStructure (p. 143)": enum,
"programDateTime (p. 143)": enum,
"baseUrl (p. 143)": "string",
"encryption (p. 143)": { 
  "initializationVectorInManifest (p. 140)": enum,
  "constantInitializationVector (p. 140)": "string",
  "staticKeyProvider (p. 140)": { 
    "keyFormatVersions (p. 189)": "string",
    "keyFormat (p. 189)": "string",
    "staticKeyValue (p. 190)": "string",
    "url (p. 190)": "string"
  },
  "type (p. 140)": enum,
  "encryptionMethod (p. 140)": enum,
  "spekeKeyProvider (p. 141)": { 
    "resourceId (p. 189)": "string",
    "systemIds (p. 189)": [ 
      "string"
    ],
    "url (p. 189)": "string"
  }
},
"adMarkers (p. 143)": [ enum ],
"programDateTimePeriod (p. 143)": integer,
"manifestCompression (p. 143)": enum,
"segmentLength (p. 143)": integer,
"manifestDurationFormat (p. 144)": enum
},
"name (p. 182)": "string",
"customName (p. 183)": "string"
],
"nielsenConfiguration (p. 153)": { 
  "distributorId (p. 179)": "string",
  "breakoutCode (p. 179)": integer
},
"inputs (p. 154)": [ 
  { 
    "audioSelectors (p. 148)": [ 
    ],
    "audioSelectorGroups (p. 148)": { 
    },
    "filterEnable (p. 148)": enum,
    "deblockFilter (p. 148)": enum,
    "videoSelector (p. 148)": { 
      "colorSpace (p. 197)": enum,
      "hdr10Metadata (p. 197)": { 
        "redPrimaryY (p. 136)": integer,
        "greenPrimaryY (p. 136)": integer,
        "whitePointX (p. 136)": integer,
        "maxLuminance (p. 137)": integer,
        "greenPrimaryX (p. 137)": integer,
        "whitePointY (p. 137)": integer,
"redPrimaryX (p. 137)": integer,
"bluePrimaryX (p. 137)": integer,
"bluePrimaryY (p. 138)": integer,
"maxFrameAverageLightLevel (p. 138)": integer,
"maxContentLightLevel (p. 138)": integer,
"minLuminance (p. 138)": integer,
"programNumber (p. 197)": integer,
"pid (p. 197)": integer,
"colorSpaceUsage (p. 197)": enum,
"filterStrength (p. 148)": integer,
"programNumber (p. 148)": integer,
"timecodeSource (p. 149)": enum,
"denoiseFilter (p. 149)": enum,
"captionSelectors (p. 149)": {
},
"inputClippings (p. 149)": [
  {
    "startTimecode (p. 146)": "string",
    "endTimecode (p. 147)": "string"
  }
],
"psiControl (p. 149)": enum,
"timedMetadataInsertion (p. 154)": {
  "id3Insertions (p. 192)": [
    {
      "id3 (p. 146)": "string",
      "timecode (p. 146)": "string"
    }
  ],
  "availBlanking (p. 154)": {
    "availBlankingImage (p. 88)": "string"
  }
},
"name (p. 152)": "string",
"description (p. 152)": "string",
"arn (p. 152)": "string",
"category (p. 152)": "string",
"type (p. 152)": enum,
"queue (p. 153)": "string"
}
]

Example CreateJobTemplateResponse

{
  "jobTemplate (p. 102)": {
    "createdAt (p. 151)": "string",
    "lastUpdated (p. 152)": "string",
    "settings (p. 152)": {
      "timecodeConfig (p. 153)": {
        "timestampOffset (p. 191)": "string",
        "anchor (p. 191)": "string",
        "start (p. 192)": "string",
        "source (p. 192)": enum
      },
      "adAvailOffset (p. 153)": integer,
      "outputGroups (p. 153)": [
        
      ]
  }
}
"outputs (p. 182)": [
  {
    "extension (p. 181)": "string",
    "videoDescription (p. 181)": {
      "fixedAfd (p. 194)": integer,
      "scalingBehavior (p. 194)": enum,
      "respondToAfd (p. 194)": enum,
      "codecSettings (p. 194)": {
        "h265Settings (p. 193)": {
          "slices (p. 130)": integer,
          "minIInterval (p. 130)": integer,
          "parNumerator (p. 130)": integer,
          "flickerAdaptiveQuantization (p. 131)": enum,
          "gopSizeUnits (p. 131)": enum,
          "hrdBufferSize (p. 131)": integer,
          "qualityTuningLevel (p. 131)": enum,
          "maxBitrate (p. 131)": integer,
          "bitrate (p. 131)": integer,
          "spatialAdaptiveQuantization (p. 131)": enum,
          "sampleAdaptiveOffsetFilterMode (p. 131)": enum,
          "temporalIds (p. 132)": enum,
          "slowPal (p. 132)": enum,
          "tiles (p. 132)": enum,
          "writeMp4PackagingType (p. 132)": enum,
          "codecProfile (p. 132)": enum,
          "alternateTransferFunctionSei (p. 132)": enum,
          "unregisteredSeiTimecode (p. 132)": enum,
          "framerateControl (p. 132)": enum,
          "telecine (p. 132)": enum,
          "codecLevel (p. 132)": enum,
          "framerateConversionAlgorithm (p. 133)": enum,
          "numberReferenceFrames (p. 133)": integer,
          "temporalAdaptiveQuantization (p. 133)": enum,
          "hrdBuffeInitialFillPercentage (p. 133)": integer,
          "framerateNumerator (p. 133)": integer,
          "gopClosedCadence (p. 133)": integer,
          "numberBFramesBetweenReferenceFrames (p. 133)": integer,
          "framerateDenominator (p. 134)": integer,
          "adaptiveQuantization (p. 134)": enum,
          "interlaceMode (p. 134)": enum,
          "gopSize (p. 134)": number,
          "gopBReference (p. 134)": enum,
          "parDenominator (p. 134)": integer,
          "sceneChangeDetect (p. 134)": enum,
          "parControl (p. 135)": enum,
          "rateControlMode (p. 135)": enum
        },
        "proresSettings (p. 193)": {
          "framerateDenominator (p. 185)": integer,
          "slowPal (p. 185)": enum,
          "framerateControl (p. 185)": enum,
          "telecine (p. 185)": enum,
          "framerateConversionAlgorithm (p. 185)": enum,
          "interlaceMode (p. 186)": enum,
          "parNumerator (p. 186)": integer,
          "codecProfile (p. 186)": enum,
          "parDenominator (p. 186)": integer,
          "parControl (p. 186)": integer,
          "framerateNumerator (p. 186)": integer
        },
        "mpeg2Settings (p. 193)": {
          "minIInterval (p. 173)": integer,
          "parNumerator (p. 173)": integer,
          "gopSizeUnits (p. 173)": enum,
          "hrdBuffeSize (p. 174)": integer,
        }
      }
    }
  }
]
"qualityTuningLevel (p. 174)": enum,
"maxBitrate (p. 174)": integer,
"bitrate (p. 174)": integer,
"spatialAdaptiveQuantization (p. 174)": enum,
"slowPal (p. 174)": enum,
"intraDCPrecision (p. 174)": enum,
"codecProfile (p. 174)": enum,
"softness (p. 175)": integer,
"framerateControl (p. 175)": enum,
"telecine (p. 175)": enum,
"codecLevel (p. 175)": enum,
"framerateConversionAlgorithm (p. 175)": enum,
"temporalAdaptiveQuantization (p. 175)": enum,
"hrdBufferInitialFillPercentage (p. 175)": integer,
"framerateDenominator (p. 175)": integer,
"gopClosedCadence (p. 176)": integer,
"numberBFramesBetweenReferenceFrames (p. 176)": integer,
"framerateNumerator (p. 176)": integer,
"syntax (p. 176)": enum,
"parDenominator (p. 177)": integer,
"sceneChangeDetect (p. 177)": enum,
"parControl (p. 177)": enum,
"rateControlMode (p. 177)": enum
},
"frameCaptureSettings (p. 194)": {
  "framerateDenominator (p. 118)": integer,
  "maxCaptures (p. 118)": integer,
  "framerateNumerator (p. 118)": integer,
  "quality (p. 118)": integer
},
"h264Settings (p. 194)": {
  "slices (p. 121)": integer,
  "minIInterval (p. 122)": integer,
  "parNumerator (p. 122)": integer,
  "flickerAdaptiveQuantization (p. 122)": enum,
  "gopSizeUnits (p. 122)": enum,
  "hrdBufferSize (p. 122)": integer,
  "qualityTuningLevel (p. 122)": enum,
  "maxBitrate (p. 122)": integer,
  "bitrate (p. 123)": integer,
  "spatialAdaptiveQuantization (p. 123)": enum,
  "slowPal (p. 123)": enum,
  "codecProfile (p. 123)": enum,
  "unregisteredSeiTimecode (p. 123)": enum,
  "softness (p. 123)": integer,
  "framerateControl (p. 123)": enum,
  "telecine (p. 123)": enum,
  "codecLevel (p. 124)": enum,
  "framerateConversionAlgorithm (p. 124)": enum,
  "numberReferenceFrames (p. 124)": integer,
  "temporalAdaptiveQuantization (p. 124)": enum,
  "repeatPps (p. 124)": integer,
  "hrdBufferInitialFillPercentage (p. 124)": integer,
  "framerateNumerator (p. 124)": integer,
  "gopClosedCadence (p. 124)": integer,
  "numberBFramesBetweenReferenceFrames (p. 125)": integer,
  "framerateDenominator (p. 125)": integer,
  "entropyEncoding (p. 125)": enum,
  "fieldEncoding (p. 125)": enum,
  "adaptiveQuantization (p. 125)": enum,
  "interlaceMode (p. 125)": enum,
  "gopSize (p. 125)": number,
  "gopBReference (p. 126)": enum,
"syntax (p. 126)": enum,
"parDenominator (p. 126)": integer,
"sceneChangeDetect (p. 126)": enum,
"parControl (p. 126)": enum,
"rateControlMode (p. 126)": enum
}
,
"afdsSignaling (p. 194)": enum,
"colorMetadata (p. 194)": enum,
"timecodeInsertion (p. 194)": enum,
"width (p. 195)": integer,
"videoPreprocessors (p. 195)": {
    "noiseReducer (p. 196)": {
        "filter (p. 179)": enum,
        "filterSettings (p. 179)": {
            "strength (p. 180)": integer
        }
    },
    "spatialFilterSettings (p. 179)": {
        "strength (p. 180)": integer,
        "postFilterSharpenStrength (p. 180)": integer,
        "speed (p. 180)": integer
    }
}
,
"timecodeBurnin (p. 196)": {
    "prefix (p. 190)": "string",
    "fontSize (p. 191)": integer,
    "position (p. 191)": enum
}
,
"colorCorrector (p. 196)": {
    "saturation (p. 99)": integer,
    "brightness (p. 99)": integer,
    "hdr10Metadata (p. 99)": {
        "redPrimaryY (p. 136)": integer,
        "greenPrimaryY (p. 136)": integer,
        "whitePointX (p. 136)": integer,
        "maxLuminance (p. 137)": integer,
        "greenPrimaryX (p. 137)": integer,
        "whitePointY (p. 137)": integer,
        "redPrimaryX (p. 137)": integer,
        "bluePrimaryX (p. 137)": integer,
        "bluePrimaryY (p. 138)": integer,
        "maxFrameAverageLightLevel (p. 138)": integer,
        "maxContentLightLevel (p. 138)": integer,
        "minLuminance (p. 138)": integer
    },
    "contrast (p. 100)": integer,
    "hue (p. 100)": integer,
    "colorSpaceConversion (p. 100)": enum
}
,
"deinterlacer (p. 196)": {
    "mode (p. 105)": enum,
    "control (p. 105)": enum,
    "algorithm (p. 105)": enum
}
,
"imageInserter (p. 196)": {
    "insertableImages (p. 146)": [
        {
            "duration (p. 149)": integer,
            "fadeOut (p. 150)": integer,
            "imageY (p. 150)": integer,
            "fadeIn (p. 150)": integer,
            "imageX (p. 150)": integer,
            "width (p. 150)": integer,
            "startTime (p. 151)": "string",
            "opacity (p. 151)": integer,
            "layer (p. 151)": integer
        }
    ]
}
"bitrate (p. 82)": integer,
"lfeFilter (p. 83)": enum,
"bitstreamMode (p. 83)": enum,
"sampleRate (p. 83)": integer
},
"aiffSettings (p. 85)": {
"channels (p. 83)": integer,
"bitDepth (p. 83)": integer,
"sampleRate (p. 83)": integer
},
"eac3Settings (p. 85)": {
"dialnorm (p. 113)": integer,
"passthroughControl (p. 113)": enum,
"metadataControl (p. 113)": enum,
"bitrate (p. 113)": integer,
"dynamicRangeCompressionRf (p. 113)": enum,
"sampleRate (p. 113)": integer,
"surroundExMode (p. 113)": enum,
"ltRtSurroundMixLevel (p. 114)": number,
"dynamicRangeCompressionLine (p. 114)": enum,
"lfeControl (p. 114)": enum,
"codingMode (p. 114)": enum,
"surroundMode (p. 114)": enum,
"attenuationControl (p. 114)": enum,
"lfeFilter (p. 114)": enum,
"phaseControl (p. 114)": enum,
"ltRtCenterMixLevel (p. 114)": number,
"dcFilter (p. 115)": enum,
"bitstreamMode (p. 115)": enum,
"stereoDownmix (p. 115)": enum,
"loRoSurroundMixLevel (p. 115)": number,
"loRoCenterMixLevel (p. 115)": number
},
"mp2Settings (p. 85)": {
"channels (p. 169)": integer,
"bitrate (p. 170)": integer,
"sampleRate (p. 170)": integer
},
"languageCode (p. 86)": enum,
"streamName (p. 86)": "string",
"audioNormalizationSettings (p. 86)": {
"targetLkfs (p. 87)": number,
"algorithmControl (p. 88)": enum,
"loudnessLogging (p. 88)": enum,
"peakCalculation (p. 88)": enum,
"correctionGateLevel (p. 88)": integer,
"algorithm (p. 88)": enum
},
"customLanguageCode (p. 86)": "string"
],
"containerSettings (p. 181)": {
"container (p. 101)": enum,
"mp4Settings (p. 101)": {
"mp4MajorBrand (p. 170)": "string",
"moovPlacement (p. 171)": enum,
"cslgAtom (p. 171)": enum,
"freeSpaceBox (p. 171)": enum
},
"m3u8Settings (p. 101)": {
"pmtPid (p. 166)": integer,
"nielsenId3 (p. 166)": enum,
"pcrPid (p. 166)": integer,
"audioPids (p. 166)": [ integer
Schemas

{,
  "audioFramesPerPes (p. 166)": integer,
  "scte35Pid (p. 166)": integer,
  "transportStreamId (p. 167)": integer,
  "pcrControl (p. 167)": enum,
  "videoPid (p. 167)": integer,
  "privateMetadataPid (p. 167)": integer,
  "pmtInterval (p. 167)": integer,
  "patInterval (p. 167)": integer,
  "programNumber (p. 168)": integer,
  "timedMetadataPid (p. 168)": integer,
  "timedMetadata (p. 168)": enum,
  "scte35Source (p. 168)": enum
},

"m2tsSettings (p. 101)": {
  "dvbTeletextPid (p. 160)": integer,
  "nielsenId3 (p. 161)": enum,
  "bitrate (p. 161)": integer,
  "segmentationTime (p. 161)": number,
  "audioPids (p. 161)": [
    integer
  ],
  "rateMode (p. 161)": enum,
  "maxPcrInterval (p. 161)": integer,
  "audioFramesPerPes (p. 161)": integer,
  "ebpAudioInterval (p. 162)": enum,
  "fragmentTime (p. 162)": number,
  "scte35Pid (p. 162)": integer,
  "pmtInterval (p. 162)": integer,
  "privateMetadataPid (p. 162)": integer,
  "segmentationStyle (p. 162)": enum,
  "audioBufferModel (p. 162)": enum,
  "programNumber (p. 163)": integer,
  "dvbNitSettings (p. 163)": {
    "networkName (p. 105)": "string",
    "networkId (p. 106)": integer,
    "nitInterval (p. 106)": integer
  },
  "timedMetadataPid (p. 163)": integer,
  "scte35Source (p. 163)": enum,
  "pmtPid (p. 163)": integer,
  "bufferModel (p. 163)": enum,
  "ebpPlacement (p. 163)": enum,
  "dvbSdtSettings (p. 163)": {
    "sdtInterval (p. 106)": integer,
    "serviceName (p. 106)": "string",
    "serviceProviderName (p. 106)": "string",
    "outputSdt (p. 106)": enum
  },
  "nullPacketBitrate (p. 164)": number,
  "pcrPid (p. 164)": integer,
  "minEbpInterval (p. 164)": integer,
  "transportStreamId (p. 164)": integer,
  "videoPid (p. 164)": integer,
  "pcrControl (p. 165)": enum,
  "esRateInPes (p. 165)": enum,
  "segmentationMarkers (p. 165)": enum,
  "dvbTdtSettings (p. 165)": {
    "tdtInterval (p. 111)": integer
  },
  "patInterval (p. 165)": integer,
  "dvbSubPids (p. 165)": [
    integer
  ]
},

"movSettings (p. 101)": {
  ""
Schemas

"reference (p. 169)" : enum,
"paddingControl (p. 169)" : enum,
"mpeg2FourCCControl (p. 169)" : enum,
"cs1gAtom (p. 169)" : enum,
"clapAtom (p. 169)" : enum
},
"f4vSettings (p. 101)" : {
  "moovPlacement (p. 117)" : enum
}
},
"preset (p. 181)" : "string",
"outputSettings (p. 182)" : {
  "hlsSettings (p. 184)" : {
    "iFrameOnlyManifest (p. 145)" : enum,
    "audioRenditionSets (p. 145)" : "string",
    "audioTrackType (p. 145)" : enum,
    "segmentModifier (p. 145)" : "string",
    "audioGroupId (p. 145)" : "string"
  }
},
"captionDescriptions (p. 182)" : [
  {
    "captionSelectorName (p. 92)" : "string",
    "languageDescription (p. 93)" : "string",
    "destinationSettings (p. 93)" : {
      "burninDestinationSettings (p. 93)" : {
        "xPosition (p. 89)" : integer,
        "backgroundColor (p. 89)" : enum,
        "teletextSpacing (p. 89)" : enum,
        "yPosition (p. 89)" : integer,
        "backgroundColor (p. 89)" : integer,
        "fontOpacity (p. 90)" : integer,
        "fontResolution (p. 90)" : integer,
        "shadowOpacity (p. 90)" : integer,
        "shadowYOffset (p. 90)" : integer,
        "outlineSize (p. 90)" : integer,
        "outlineColor (p. 90)" : enum,
        "fontSize (p. 91)" : integer,
        "shadowXOffset (p. 91)" : integer,
        "alignment (p. 91)" : enum,
        "shadowColor (p. 91)" : enum,
        "fontColor (p. 91)" : enum
      },
      "teletextDestinationSettings (p. 93)" : {
        "pageNumber (p. 190)" : "string"
      },
      "ttmlDestinationSettings (p. 93)" : {
        "stylePassthrough (p. 193)" : enum
      },
      "destinationType (p. 93)" : enum,
      "dvbSubDestinationSettings (p. 94)" : {
        "xPosition (p. 107)" : integer,
        "backgroundColor (p. 107)" : enum,
        "teletextSpacing (p. 107)" : enum,
        "yPosition (p. 107)" : integer,
        "backgroundColor (p. 107)" : integer,
        "fontOpacity (p. 107)" : integer,
        "fontResolution (p. 108)" : integer,
        "shadowOpacity (p. 108)" : integer,
        "shadowYOffset (p. 108)" : integer,
        "outlineSize (p. 108)" : integer,
        "outlineColor (p. 108)" : enum,
        "fontSize (p. 108)" : integer,
        "shadowXOffset (p. 109)" : integer,
        "alignment (p. 109)" : enum,
        "shadowColor (p. 109)" : enum,
"fontColor (p. 109)": enum
},
"sccDestinationSettings (p. 94)": {
  "framerate (p. 189)": enum
}
},
"languageCode (p. 93)": enum,
"customLanguageCode (p. 93)": "string"
}
],
"nameModifier (p. 182)": "string"
}
],
"outputGroupSettings (p. 182)": {
  "dashIsoGroupSettings (p. 183)": {
    "fragmentLength (p. 103)": integer,
    "baseUrl (p. 103)": "string",
    "minBufferTime (p. 103)": integer,
    "encryption (p. 103)": {
      "spekeKeyProvider (p. 103)": {
        "resourceId (p. 189)": "string",
        "systemIds (p. 189)": [
          "string"
        ],
        "url (p. 189)": "string"
      }
    },
    "destination (p. 103)": "string",
    "segmentLength (p. 104)": integer,
    "segmentControl (p. 104)": enum,
    "hbbtvCompliance (p. 104)": enum
  },
  "fileGroupSettings (p. 183)": {
    "destination (p. 117)": "string"
  },
  "msSmoothGroupSettings (p. 183)": {
    "fragmentLength (p. 178)": integer,
    "encryption (p. 178)": {
      "spekeKeyProvider (p. 178)": {
        "resourceId (p. 189)": "string",
        "systemIds (p. 189)": [
          "string"
        ],
        "url (p. 189)": "string"
      }
    },
    "audioDeduplication (p. 178)": enum,
    "manifestEncoding (p. 178)": enum,
    "destination (p. 179)": "string"
  },
  "cmafGroupSettings (p. 183)": {
    "writeHlsManifest (p. 96)": enum,
    "writeDashManifest (p. 96)": enum,
    "fragmentLength (p. 96)": integer,
    "streamInfResolution (p. 97)": enum,
    "minBufferTime (p. 97)": integer,
    "clientCache (p. 97)": enum,
    "codecSpecification (p. 97)": enum,
    "destination (p. 97)": "string",
    "segmentControl (p. 97)": enum,
    "baseUrl (p. 97)": "string",
    "encryption (p. 97)": {
      "initializationVectorInManifest (p. 95)": enum,
      "constantInitializationVector (p. 96)": "string",
      "staticKeyProvider (p. 96)": {
        "keyFormatVersions (p. 189)": "string",
        "systemIds (p. 189)": [
          "string"
        ],
        "url (p. 189)": "string"
      }
    }
  }
}
AWS Elemental MediaConvert API Reference API Reference
Schemas
"keyFormat (p. 189)": "string",
"staticKeyValue (p. 190)": "string",
"url (p. 190)": "string"

},
"type (p. 96)": enum,
"encryptionMethod (p. 96)": enum

},
"manifestCompression (p. 98)": enum,
"segmentLength (p. 98)": integer,
"manifestDurationFormat (p. 98)": enum

},
"type (p. 183)": enum,
"hlsGroupSettings (p. 183)": {
"segmentsPerSubdirectory (p. 141)": integer,
"streamInfResolution (p. 141)": enum,
"timestampDeltaMilliseconds (p. 141)": integer,
"outputSelection (p. 141)": enum,
"captionLanguageMappings (p. 141)": [
{
"languageDescription (p. 139)": "string",
"captionChannel (p. 139)": integer,
"languageCode (p. 139)": enum,
"customLanguageCode (p. 139)": "string"
}
],
"clientCache (p. 142)": enum,
"codecSpecification (p. 142)": enum,
"destination (p. 142)": "string",
"timedMetadataId3Frame (p. 142)": enum,
"segmentControl (p. 142)": enum,
"timedMetadataId3Period (p. 142)": integer,
"captionLanguageSetting (p. 142)": enum,
"minSegmentLength (p. 142)": integer,
"directoryStructure (p. 143)": enum,
"programDateTime (p. 143)": enum,
"baseUrl (p. 143)": "string",
"encryption (p. 143)": {
"initializationVectorInManifest (p. 140)": enum,
"constantInitializationVector (p. 140)": "string",
"staticKeyProvider (p. 140)": {
"keyFormatVersions (p. 189)": "string",
"keyFormat (p. 189)": "string",
"staticKeyValue (p. 190)": "string",
"url (p. 190)": "string"
},
"type (p. 140)": enum,
"encryptionMethod (p. 140)": enum,
"spekeKeyProvider (p. 141)": {
"resourceId (p. 189)": "string",
"systemIds (p. 189)": [
"string"
],
"url (p. 189)": "string"
}
},
"adMarkers (p. 143)": [
enum
],
"programDateTimePeriod (p. 143)": integer,
"manifestCompression (p. 143)": enum,
"segmentLength (p. 143)": integer,
"manifestDurationFormat (p. 144)": enum
}

},
"name (p. 182)": "string",
"customName (p. 183)": "string"

77


{  
  "nielsenConfiguration (p. 153)": {  
    "distributorId (p. 179)": "string",  
    "breakoutCode (p. 179)": integer  
  },  
  "inputs (p. 154)": [  
    {  
      "audioSelectors (p. 148)": {  
      },  
      "audioSelectorGroups (p. 148)": {  
      },  
      "filterEnable (p. 148)": enum,  
      "deblockFilter (p. 148)": enum,  
      "videoSelector (p. 148)": {  
        "colorSpace (p. 197)": enum,  
        "hdr10Metadata (p. 197)": {  
          "redPrimaryY (p. 136)": integer,  
          "greenPrimaryY (p. 136)": integer,  
          "whitePointX (p. 136)": integer,  
          "maxLuminance (p. 137)": integer,  
          "greenPrimaryX (p. 137)": integer,  
          "whitePointY (p. 137)": integer,  
          "redPrimaryX (p. 137)": integer,  
          "bluePrimaryX (p. 138)": integer,  
          "bluePrimaryY (p. 138)": integer,  
          "maxFrameAverageLightLevel (p. 138)": integer,  
          "maxContentLightLevel (p. 138)": integer,  
          "minLuminance (p. 138)": integer  
        },  
        "programNumber (p. 197)": integer,  
        "pid (p. 197)": integer,  
        "colorSpaceUsage (p. 197)": enum  
      },  
      "filterStrength (p. 148)": integer,  
      "programNumber (p. 148)": integer,  
      "timecodeSource (p. 149)": enum,  
      "denoiseFilter (p. 149)": enum,  
      "captionSelectors (p. 149)": {  
      },  
      "inputClippings (p. 149)": [  
        {  
          "startTimecode (p. 146)": "string",  
          "endTimecode (p. 147)": "string"  
        }  
      ],  
      "psiControl (p. 149)": enum  
    },  
    {  
      "timedMetadataInsertion (p. 154)": {  
        "id3Insertions (p. 192)": [  
          {  
            "id3 (p. 146)": "string",  
            "timecode (p. 146)": "string"  
          }  
        ],  
        "availBlanking (p. 154)": {  
          "availBlankingImage (p. 88)": "string"  
        }  
      },  
      "name (p. 152)": "string",  
      "description (p. 152)": "string",  
      "arn (p. 152)": "string",  
      "category (p. 152)": "string",  
      "type (p. 152)": enum  
    }  
  ]}
"queue (p. 153)": "string"
}

**Example ExceptionBody**

```
{
"message (p. 116)": "string"
}
```

**Properties**

**AacAudioDescriptionBroadcasterMix (enum)**

Choose BROADCASTER_MIXED_AD when the input contains pre-mixed main audio + audio description (AD) as a stereo pair. The value for AudioType will be set to 3, which signals to downstream systems that this stream contains "broadcaster mixed AD". Note that the input received by the encoder must contain pre-mixed audio; the encoder does not perform the mixing. When you choose BROADCASTER_MIXED_AD, the encoder ignores any values you provide in AudioType and FollowInputAudioType. Choose NORMAL when the input does not contain pre-mixed audio + audio description (AD). In this case, the encoder will use any values you provide for AudioType and FollowInputAudioType.

- BROADCASTER_MIXED_AD
- NORMAL

**AacCodecProfile (enum)**

AAC Profile.

- LC
- HEV1
- HEV2

**AacCodingMode (enum)**

Mono (Audio Description), Mono, Stereo, or 5.1 channel layout. Valid values depend on rate control mode and profile. "1.0 - Audio Description (Receiver Mix)" setting receives a stereo description plus control track and emits a mono AAC encode of the description track, with control data emitted in the PES header as per ETSI TS 101 154 Annex E.

- AD_RECEIVER_MIX
- CODING_MODE_1_0
- CODING_MODE_1_1
- CODING_MODE_2_0
- CODING_MODE_5_1

**AacRateControlMode (enum)**

Rate Control Mode.

- CBR
- VBR
**AacRawFormat (enum)**

Enables LATM/LOAS AAC output. Note that if you use LATM/LOAS AAC in an output, you must choose "No container" for the output container.

- LATM_LOAS
- NONE

**AacSettings**

**vbrQuality**

  - Type: string
  - Required: False

**codecProfile**

  - Type: string
  - Required: False

**codingMode**

  - Type: string
  - Required: True

**specification**

  - Type: string
  - Required: False

**bitrate**

Average bitrate in bits/second. Defaults and valid values depend on rate control mode and profile.

- Type: integer
- Required: False
- Minimum: 6000
- Maximum: 1024000

**rawFormat**

  - Type: string
  - Required: False

**rateControlMode**

  - Type: string
  - Required: False

**sampleRate**

Sample rate in Hz. Valid values depend on rate control mode and profile.
**Properties**

**Type**: integer
**Required**: True
**Minimum**: 8000
**Maximum**: 96000

**audioDescriptionBroadcasterMix**

**Type**: string
**Required**: False

**AacSpecification (enum)**

Use MPEG-2 AAC instead of MPEG-4 AAC audio for raw or MPEG-2 Transport Stream containers.

- MPEG2
- MPEG4

**AacVbrQuality (enum)**

VBR Quality Level - Only used if rate_control_mode is VBR.

- LOW
- MEDIUM_LOW
- MEDIUM_HIGH
- HIGH

**Ac3BitstreamMode (enum)**

Specifies the "Bitstream Mode" (bsmod) for the emitted AC-3 stream. See ATSC A/52-2012 for background on these values.

- COMPLETE_MAIN
- COMMENTARY
- DIALOGUE
- EMERGENCY
- HEARING_IMPAIRED
- MUSIC_AND_EFFECTS
- VISUALLY_IMPAIRED
- VOICE_OVER

**Ac3CodingMode (enum)**

Dolby Digital coding mode. Determines number of channels.

- CODING_MODE_1_0
- CODING_MODE_1_1
- CODING_MODE_2_0
- CODING_MODE_3_2_LFE

**Ac3DynamicRangeCompressionProfile (enum)**

If set to FILM_STANDARD, adds dynamic range compression signaling to the output bitstream as defined in the Dolby Digital specification.

- FILM_STANDARD
NONE

**Ac3LfeFilter (enum)**
Applies a 120Hz lowpass filter to the LFE channel prior to encoding. Only valid with 3_2_LFE coding mode.

- ENABLED
- DISABLED

**Ac3MetadataControl (enum)**
When set to FOLLOW_INPUT, encoder metadata will be sourced from the DD, DD+, or DolbyE decoder that supplied this audio data. If audio was not supplied from one of these streams, then the static metadata settings will be used.

- FOLLOW_INPUT
- USE_CONFIGURED

**Ac3Settings**

**dynamicRangeCompressionProfile**

- Type: string
- Required: False

**dialnorm**
Sets the dialnorm for the output. If blank and input audio is Dolby Digital, dialnorm will be passed through.

- Type: integer
- Required: False
- Minimum: 1
- Maximum: 31

**codingMode**

- Type: string
- Required: False

**metadataControl**

- Type: string
- Required: False

**bitrate**
Average bitrate in bits/second. Valid bitrates depend on the coding mode.

- Type: integer
- Required: False
- Minimum: 64000
- Maximum: 640000
**IfeFilter**

- **Type:** string
- **Required:** False

**bitstreamMode**

- **Type:** string
- **Required:** False

**sampleRate**

Sample rate in hz. Sample rate is always 48000.

- **Type:** integer
- **Required:** False
- **Minimum:** 48000
- **Maximum:** 48000

**AfdSignaling (enum)**

This setting only applies to H.264 and MPEG2 outputs. Use Insert AFD signaling (AfdSignaling) to specify whether the service includes AFD values in the output video data and what those values are. * Choose None to remove all AFD values from this output. * Choose Fixed to ignore input AFD values and instead encode the value specified in the job. * Choose Auto to calculate output AFD values based on the input AFD scaler data.

- **NONE**
- **AUTO**
- **FIXED**

**AiffSettings**

**channels**

Set Channels to specify the number of channels in this output audio track. Choosing Mono in the console will give you 1 output channel; choosing Stereo will give you 2. In the API, valid values are 1 and 2.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 2

**bitDepth**

Specify Bit depth (BitDepth), in bits per sample, to choose the encoding quality for this audio track.

- **Type:** integer
- **Required:** False
- **Minimum:** 16
- **Maximum:** 24

**sampleRate**

Sample rate in hz.
**Properties**

**Type**: integer
- **Required**: False
- **Minimum**: 8000
- **Maximum**: 192000

**AncillarySourceSettings**

**sourceAncillaryChannelNumber**

Specifies the 608 channel number in the ancillary data track from which to extract captions. Unused for passthrough.

- **Type**: integer
  - **Required**: False
  - **Minimum**: 1
  - **Maximum**: 4

**AntiAlias (enum)**

Enable Anti-alias (AntiAlias) to enhance sharp edges in video output when your input resolution is much larger than your output resolution. Default is enabled.

- DISABLED
- ENABLED

**AudioCodec (enum)**

Type of Audio codec.

- AAC
- MP2
- WAV
- AIFF
- AC3
- EAC3
- PASSTHROUGH

**AudioCodecSettings**

**codec**

- **Type**: string
  - **Required**: True

**wavSettings**

- **Type**: WavSettings (p. 198)
  - **Required**: False

**aacSettings**

- **Type**: AacSettings (p. 80)
  - **Required**: False
ac3Settings
Type: Ac3Settings (p. 82)
Required: False

aiffSettings
Type: AiffSettings (p. 83)
Required: False

eac3Settings
Type: Eac3Settings (p. 113)
Required: False

mp2Settings
Type: Mp2Settings (p. 169)
Required: False

AudioDefaultSelection (enum)
Enable this setting on one audio selector to set it as the default for the job. The service uses this default for outputs where it can't find the specified input audio. If you don't set a default, those outputs have no audio.

DEFAULT
NOT_DEFAULT

AudioDescription

audioTypeControl
Type: string
Required: False

languageCodeControl
Type: string
Required: False

remixSettings
Advanced audio remixing settings.

Type: RemixSettings (p. 187)
Required: False

audioType
Applies only if Follow Input Audio Type is unchecked (false). A number between 0 and 255. The following are defined in ISO-IEC 13818-1: 0 = Undefined, 1 = Clean Effects, 2 = Hearing Impaired, 3 = Visually Impaired Commentary, 4-255 = Reserved.
Properties

audioSourceName

Specifies which audio data to use from each input. In the simplest case, specify an "Audio Selector": #inputs-audio_selector by name based on its order within each input. For example if you specify "Audio Selector 3", then the third audio selector will be used from each input. If an input does not have an "Audio Selector 3", then the audio selector marked as "default" in that input will be used. If there is no audio selector marked as "default", silence will be inserted for the duration of that input. Alternatively, an "Audio Selector Group": #inputs-audio_selector_group name may be specified, with similar default/silence behavior. If no audio_source_name is specified, then "Audio Selector 1" will be chosen automatically.

Type: string
Required: False

codecSettings

Type: AudioCodecSettings (p. 84)
Required: True

languageCode

Indicates the language of the audio output track. The ISO 639 language specified in the 'Language Code' drop down will be used when 'Follow Input Language Code' is not selected or when 'Follow Input Language Code' is selected but there is no ISO 639 language code specified by the input.

Type: string
Required: False

streamName

Used for MS Smooth and Apple HLS outputs. Indicates the name displayed by the player (eg. English, or Director Commentary). Alphanumeric characters, spaces, and underscore are legal.

Type: string
Required: False
Pattern: ^[\w\s]*$

audioNormalizationSettings

Type: AudioNormalizationSettings (p. 87)
Required: False

customLanguageCode

Specify the language for this audio output track, using the ISO 639-2 or ISO 639-3 three-letter language code. The language specified will be used when 'Follow Input Language Code' is not selected or when 'Follow Input Language Code' is selected but there is no ISO 639 language code specified by the input.

Type: string
Required: False
Pattern: ^[A-Za-z]{3}$

**AudioLanguageCodeControl (enum)**
Choosing FOLLOW_INPUT will cause the ISO 639 language code of the output to follow the ISO 639 language code of the input. The language specified for languageCode' will be used when USE_CONFIGURED is selected or when FOLLOW_INPUT is selected but there is no ISO 639 language code specified by the input.

- FOLLOW_INPUT
- USE_CONFIGURED

**AudioNormalizationAlgorithm (enum)**
Audio normalization algorithm to use. 1770-1 conforms to the CALM Act specification, 1770-2 conforms to the EBU R-128 specification.

- ITU_BS_1770_1
- ITU_BS_1770_2

**AudioNormalizationAlgorithmControl (enum)**
When enabled the output audio is corrected using the chosen algorithm. If disabled, the audio will be measured but not adjusted.

- CORRECT_AUDIO
- MEASURE_ONLY

**AudioNormalizationLoudnessLogging (enum)**
If set to LOG, log each output's audio track loudness to a CSV file.

- LOG
- DONT_LOG

**AudioNormalizationPeakCalculation (enum)**
If set to TRUE_PEAK, calculate and log the TruePeak for each output's audio track loudness.

- TRUE_PEAK
- NONE

**AudioNormalizationSettings**

**targetLkfs**
Target LKFS(loudness) to adjust volume to. If no value is entered, a default value will be used according to the chosen algorithm. The CALM Act (1770-1) recommends a target of -24 LKFS. The EBU R-128 specification (1770-2) recommends a target of -23 LKFS.

- Type: number
- Required: False
- Format: float
- Minimum: -59.0
- Maximum: 0.0
algorithmControl

Type: string
Required: False

loudnessLogging

Type: string
Required: False

peakCalculation

Type: string
Required: False

correctionGateLevel

Content measuring above this level will be corrected to the target level. Content measuring below this level will not be corrected. Gating only applies when not using real_time_correction.

Type: integer
Required: False
Minimum: -70
Maximum: 0

algorithm

Type: string
Required: False

AudioSelectorType (enum)

Specifies the type of the audio selector.

PID
TRACK
LANGUAGE_CODE

AudioTypeControl (enum)

When set to FOLLOW_INPUT, if the input contains an ISO 639 audio_type, then that value is passed through to the output. If the input contains no ISO 639 audio_type, the value in Audio Type is included in the output. Otherwise the value in Audio Type is included in the output. Note that this field and audioType are both ignored if audioDescriptionBroadcasterMix is set to BROADCASTER_MIXED_AD.

FOLLOW_INPUT
USE_CONFIGURED

AvailBlanking

availBlankingImage

Blanking image to be used. Leave empty for solid black. Only bmp and png images are supported.
Properties

**Type**
- **string**
- **Required**: False
- **Pattern**: `^(s3://)(.*\./\.(.*bmp|bmp|png|PNG)$`

**BurninDestinationSettings**

**xPosition**

Specifies the horizontal position of the caption relative to the left side of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the left of the output. If no explicit x_position is provided, the horizontal caption position will be determined by the alignment parameter. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 2147483647

**backgroundColor**

- **Type**: string
- **Required**: False

**teletextSpacing**

- **Type**: string
- **Required**: False

**yPosition**

Specifies the vertical position of the caption relative to the top of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the top of the output. If no explicit y_position is provided, the caption will be positioned towards the bottom of the output. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 2147483647

**backgroundOpacity**

Specifies the opacity of the background rectangle. 255 is opaque; 0 is transparent. Leaving this parameter blank is equivalent to setting it to 0 (transparent). All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 255
**fontOpacity**

Specifies the opacity of the burned-in captions. 255 is opaque; 0 is transparent. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: True
- **Minimum**: 0
- **Maximum**: 255

**fontResolution**

Font resolution in DPI (dots per inch); default is 96 dpi. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: False
- **Minimum**: 96
- **Maximum**: 600

**shadowOpacity**

Specifies the opacity of the shadow. 255 is opaque; 0 is transparent. Leaving this parameter blank is equivalent to setting it to 0 (transparent). All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 255

**shadowYOffset**

Specifies the vertical offset of the shadow relative to the captions in pixels. A value of -2 would result in a shadow offset 2 pixels above the text. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: False
- **Minimum**: -2147483648
- **Maximum**: 2147483647

**outlineSize**

Specifies font outline size in pixels. This option is not valid for source captions that are either 608/ embeded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: True
- **Minimum**: 0
- **Maximum**: 10

**outlineColor**

- **Type**: string
- **Required**: True
**Properties**

**fontSize**

A positive integer indicates the exact font size in points. Set to 0 for automatic font size selection. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 96

**shadowXOffset**

Specifies the horizontal offset of the shadow relative to the captions in pixels. A value of -2 would result in a shadow offset 2 pixels to the left. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: False
- **Minimum**: -2147483648
- **Maximum**: 2147483647

**alignment**

- **Type**: string
- **Required**: True

**shadowColor**

- **Type**: string
- **Required**: False

**fontColor**

- **Type**: string
- **Required**: False

**BurninSubtitleAlignment (enum)**

If no explicit x_position or y_position is provided, setting alignment to centered will place the captions at the bottom center of the output. Similarly, setting a left alignment will align captions to the bottom left of the output. If x and y positions are given in conjunction with the alignment parameter, the font will be justified (either left or centered) relative to those coordinates. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- CENTERED
- LEFT

**BurninSubtitleBackgroundColor (enum)**

Specifies the color of the rectangle behind the captions. All burn-in and DVB-Sub font settings must match.

- NONE
- BLACK
**WHITE**

**BurninSubtitleFontColor (enum)**

Specifies the color of the burned-in captions. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- WHITE
- BLACK
- YELLOW
- RED
- GREEN
- BLUE

**BurninSubtitleOutlineColor (enum)**

Specifies font outline color. This option is not valid for source captions that are either 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- BLACK
- WHITE
- YELLOW
- RED
- GREEN
- BLUE

**BurninSubtitleShadowColor (enum)**

Specifies the color of the shadow cast by the captions. All burn-in and DVB-Sub font settings must match.

- NONE
- BLACK
- WHITE

**BurninSubtitleTeletextSpacing (enum)**

Only applies to jobs with input captions in Teletext or STL formats. Specify whether the spacing between letters in your captions is set by the captions grid or varies depending on letter width. Choose fixed grid to conform to the spacing specified in the captions file more accurately. Choose proportional to make the text easier to read if the captions are closed caption.

- FIXED_GRID
- PROPORTIONAL

**CaptionDescription**

**captionSelectorName**

Specifies which "Caption Selector":#inputs-caption_selector to use from each input when generating captions. The name should be of the format "Caption Selector <N>", which denotes that the Nth Caption Selector will be used from each input.
Type: string  
Required: True

**languageDescription**

Human readable information to indicate captions available for players (eg. English, or Spanish). Alphanumeric characters, spaces, and underscore are legal.

Type: string  
Required: False
Pattern: ^[\w \]*$

**destinationSettings**

Type: CaptionDestinationSettings (p. 93)  
Required: True

**languageCode**

Indicates the language of the caption output track.

Type: string  
Required: False

**customLanguageCode**

Indicates the language of the caption output track, using the ISO 639-2 or ISO 639-3 three-letter language code.

Type: string  
Required: False
Pattern: ^[A-Za-z]{3}$

**CaptionDestinationSettings**

**burninDestinationSettings**

Type: BurninDestinationSettings (p. 89)  
Required: False

**teletextDestinationSettings**

Type: TeletextDestinationSettings (p. 190)  
Required: False

**ttmlDestinationSettings**

Type: TtmlDestinationSettings (p. 193)  
Required: False

**destinationType**

Type: string
Properties

Required: True

dvbSubDestinationSettings

Type: DvbSubDestinationSettings (p. 107)
Required: False

sccDestinationSettings

Type: SccDestinationSettings (p. 189)
Required: False

CaptionDestinationType (enum)

Type of Caption output, including Burn-In, Embedded, SCC, SRT, TTML, WebVTT, DVB-Sub, Teletext.

- BURN_IN
- DVB_SUB
- EMBEDDED
- SCC
- SRT
- TELETEXT
- TTML
- WEBVTT

CaptionSourceSettings

fileSourceSettings

Type: FileSourceSettings (p. 117)
Required: False

ancillarySourceSettings

Type: AncillarySourceSettings (p. 84)
Required: False

embeddedSourceSettings

Type: EmbeddedSourceSettings (p. 116)
Required: False

sourceType

Type: string
Required: True

dvbSubSourceSettings

Type: DvbSubSourceSettings (p. 109)
**Required:** False

**teletextSourceSettings**

- **Type:** TeletextSourceSettings (p. 190)
- **Required:** False

**CaptionSourceType (enum)**

Use Source (SourceType) to identify the format of your input captions. The service cannot auto-detect caption format.

ANCILLARY
DVB_SUB
EMBEDDED
SCC
TTML
STL
SRT
TELETEXT
NULL_SOURCE

**ChannelMapping**

**outputChannels**

List of output channels

- **Type:** Array of type OutputChannelMapping (p. 182)
- **Required:** True

**CmafClientCache (enum)**

When set to ENABLED, sets #EXT-X-ALLOW-CACHE:no tag, which prevents client from saving media segments for later replay.

DISABLED
ENABLED

**CmafCodecSpecification (enum)**

Specification to use (RFC-6381 or the default RFC-4281) during m3u8 playlist generation.

RFC_6381
RFC_4281

**CmafEncryptionSettings**

**initializationVectorInManifest**

- **Type:** string
- **Required:** False
constantInitializationVector
This is a 128-bit, 16-byte hex value represented by a 32-character text string. If this parameter is not set then the Initialization Vector will follow the segment number by default.

Type: string
Required: False
Pattern: ^[0-9a-fA-F]{32}$

staticKeyProvider
Type: StaticKeyProvider (p. 189)
Required: False

type
Type: string
Required: True

cryptMethod
Type: string
Required: False

CmafEncryptionType (enum)
Encrypts the segments with the given encryption scheme. Leave blank to disable. Selecting 'Disabled' in the web interface also disables encryption.

SAMPLE_AES

CmafGroupSettings
writeHlsManifest
Type: string
Required: False

writeDashManifest
Type: string
Required: False

fragmentLength
Length of fragments to generate (in seconds). Fragment length must be compatible with GOP size and Framerate. Note that fragments will end on the next keyframe after this number of seconds, so actual fragment length may be longer. When Emit Single File is checked, the fragmentation is internal to a single output file and it does not cause the creation of many output files as in other output types.

Type: integer
Required: True
Minimum: 1
Maximum: 2147483647
streamInfResolution

Type: string
Required: False

minBufferTime

Minimum time of initially buffered media that is needed to ensure smooth playout.

Type: integer
Required: False
Minimum: 0
Maximum: 2147483647

clientCache

Type: string
Required: False

codecSpecification

Type: string
Required: False

destination

Use Destination (Destination) to specify the S3 output location and the output filename base. Destination accepts format identifiers. If you do not specify the base filename in the URI, the service will use the filename of the input file. If your job has multiple inputs, the service uses the filename of the first input file.

Type: string
Required: False
Pattern: ^s3://\//

segmentControl

Type: string
Required: False

baseUrl

A partial URI prefix that will be put in the manifest file at the top level BaseURL element. Can be used if streams are delivered from a different URL than the manifest file.

Type: string
Required: False

encryption

DRM settings.

Type: CmafEncryptionSettings (p. 95)
Required: False

**manifestCompression**

Type: string
Required: False

**segmentLength**

Use this setting to specify the length, in seconds, of each individual CMAF segment. This value applies to the whole package; that is, to every output in the output group. Note that segments end on the first keyframe after this number of seconds, so the actual segment length might be slightly longer. If you set Segment control (CmafSegmentControl) to single file, the service puts the content of each output in a single file that has metadata that marks these segments. If you set it to segmented files, the service creates multiple files for each output, each with the content of one segment.

Type: integer
Required: True
Minimum: 1
Maximum: 2147483647

**manifestDurationFormat**

Type: string
Required: False

**CmafInitializationVectorInManifest (enum)**

The Initialization Vector is a 128-bit number used in conjunction with the key for encrypting blocks. If set to INCLUDE, Initialization Vector is listed in the manifest. Otherwise Initialization Vector is not in the manifest.

INCLUDE
EXCLUDE

**CmafKeyProviderType (enum)**

Indicates which type of key provider is used for encryption.

STATIC_KEY

**CmafManifestCompression (enum)**

When set to GZIP, compresses HLS playlist.

GZIP
NONE

**CmafManifestDurationFormat (enum)**

Indicates whether the output manifest should use floating point values for segment duration.

FLOATING_POINT
INTEGER
Properties

**CmafSegmentControl (enum)**
When set to SINGLE_FILE, a single output file is generated, which is internally segmented using the Fragment Length and Segment Length. When set to SEGMENTED_FILES, separate segment files will be created.

- SINGLE_FILE
- SEGMENTED_FILES

**CmafStreamInfResolution (enum)**
Include or exclude RESOLUTION attribute for video in EXT-X-STREAM-INF tag of variant manifest.

- INCLUDE
- EXCLUDE

**CmafWriteDASHManifest (enum)**
When set to ENABLED, a DASH MPD manifest will be generated for this output.

- DISABLED
- ENABLED

**CmafWriteHLSManifest (enum)**
When set to ENABLED, an Apple HLS manifest will be generated for this output.

- DISABLED
- ENABLED

**ColorCorrector**

**saturation**
Saturation level.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 100

**brightness**
Brightness level.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 100

**hdr10Metadata**

- **Type:** Hdr10Metadata (p. 136)
- **Required:** False
**contrast**

Contrast level.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 100

**hue**

Hue in degrees.

- **Type:** integer
- **Required:** False
- **Minimum:** -180
- **Maximum:** 180

**colorSpaceConversion**

- **Type:** string
- **Required:** False

**ColorMetadata (enum)**

Enable Insert color metadata (ColorMetadata) to include color metadata in this output. This setting is enabled by default.

- **IGNORE**
- **INSERT**

**ColorSpace (enum)**

If your input video has accurate color space metadata, or if you don't know about color space, leave this set to the default value FOLLOW. The service will automatically detect your input color space. If your input video has metadata indicating the wrong color space, or if your input video is missing color space metadata that should be there, specify the accurate color space here. If you choose HDR10, you can also correct inaccurate color space coefficients, using the HDR master display information controls. You must also set Color space usage (ColorSpaceUsage) to FORCE for the service to use these values.

- **FOLLOW**
- **REC_601**
- **REC_709**
- **HDR10**
- **HLG_2020**

**ColorSpaceConversion (enum)**

Determines if colorspace conversion will be performed. If set to _None_, no conversion will be performed. If _Force 601_ or _Force 709_ are selected, conversion will be performed for inputs with differing colorspaces. An input's colorspace can be specified explicitly in the "Video Selector":#inputs-video_selector if necessary.

- **NONE**
- **FORCE_601**
FORCE_709
FORCE_HDR10
FORCE_HLG_2020

**ColorSpaceUsage (enum)**

There are two sources for color metadata, the input file and the job configuration (in the Color space and HDR master display information settings). The Color space usage setting controls which takes precedence. FORCE: The system will use color metadata supplied by user, if any. If the user does not supply color metadata, the system will use data from the source. FALLBACK: The system will use color metadata from the source. If source has no color metadata, the system will use user-supplied color metadata values if available.

FORCE
FALLBACK

**ContainerSettings**

**container**

Type: string
Required: True

**mp4Settings**

Type: Mp4Settings (p. 170)
Required: False

**m3u8Settings**

Type: M3u8Settings (p. 166)
Required: False

**m2tsSettings**

Type: M2tsSettings (p. 160)
Required: False

**movSettings**

Type: MovSettings (p. 169)
Required: False

**f4vSettings**

Type: F4vSettings (p. 117)
Required: False

**ContainerType (enum)**

Container for this output. Some containers require a container settings object. If not specified, the default object will be created.
CreateJobTemplateRequest

settings

Type: JobTemplateSettings (p. 153)
Required: True

name

The name of the job template you are creating.

Type: string
Required: True

description

Optional. A description of the job template you are creating.

Type: string
Required: False

category

Optional. A category for the job template you are creating.

Type: string
Required: False

queue

Optional. The queue that jobs created from this template are assigned to. If you don't specify this, jobs will go to the default queue.

Type: string
Required: False

CreateJobTemplateResponse

jobTemplate

Type: JobTemplate (p. 151)
**Required**: False

**DashIsoEncryptionSettings**

**spekeKeyProvider**

*Type*: SpekeKeyProvider (p. 189)  
*Required*: True

**DashIsoGroupSettings**

**fragmentLength**

Length of fragments to generate (in seconds). Fragment length must be compatible with GOP size and Framerate. Note that fragments will end on the next keyframe after this number of seconds, so actual fragment length may be longer. When Emit Single File is checked, the fragmentation is internal to a single output file and it does not cause the creation of many output files as in other output types.

*Type*: integer  
*Required*: True  
*Minimum*: 1  
*Maximum*: 2147483647

**baseUrl**

A partial URI prefix that will be put in the manifest (.mpd) file at the top level BaseURL element. Can be used if streams are delivered from a different URL than the manifest file.

*Type*: string  
*Required*: False

**minBufferTime**

Minimum time of initially buffered media that is needed to ensure smooth playout.

*Type*: integer  
*Required*: False  
*Minimum*: 0  
*Maximum*: 2147483647

**encryption**

DRM settings.

*Type*: DashIsoEncryptionSettings (p. 103)  
*Required*: False

**destination**

Use Destination (Destination) to specify the S3 output location and the output filename base. Destination accepts format identifiers. If you do not specify the base filename in the URI, the service will use the filename of the input file. If your job has multiple inputs, the service uses the filename of the first input file.
Properties

Type: string
Required: False
Pattern: ^s3:/\/

segmentLength

Length of mpd segments to create (in seconds). Note that segments will end on the next keyframe after this number of seconds, so actual segment length may be longer. When Emit Single File is checked, the segmentation is internal to a single output file and it does not cause the creation of many output files as in other output types.

Type: integer
Required: True
Minimum: 1
Maximum: 2147483647

segmentControl

Type: string
Required: False

hbbtvCompliance

Type: string
Required: False

DashIsoHbbtvCompliance (enum)

Supports HbbTV specification as indicated

- HBBTV_1_5
- NONE

DashIsoSegmentControl (enum)

When set to SINGLE_FILE, a single output file is generated, which is internally segmented using the Fragment Length and Segment Length. When set to SEGMENTED_FILES, separate segment files will be created.

- SINGLE_FILE
- SEGMENTED_FILES

DeinterlaceAlgorithm (enum)

Only applies when you set Deinterlacer (DeinterlaceMode) to Deinterlace (DEINTERLACE) or Adaptive (ADAPTIVE). Motion adaptive interpolate (INTERPOLATE) produces sharper pictures, while blend (BLEND) produces smoother motion. Use (INTERPOLATE_TICKER) OR (BLEND_TICKER) if your source file includes a ticker, such as a scrolling headline at the bottom of the frame.

- INTERPOLATE
- INTERPOLATE_TICKER
- BLEND
- BLEND_TICKER
Deinterlacer

mode

Type: string
Required: False

control

Type: string
Required: False

algorithm

Type: string
Required: False

DeinterlacerControl (enum)
- When set to NORMAL (default), the deinterlacer does not convert frames that are tagged in metadata as progressive. It will only convert those that are tagged as some other type. - When set to FORCE_ALL_FRAMES, the deinterlacer converts every frame to progressive - even those that are already tagged as progressive. Turn Force mode on only if there is a good chance that the metadata has tagged frames as progressive when they are not progressive. Do not turn on otherwise; processing frames that are already progressive into progressive will probably result in lower quality video.

  FORCE_ALL_FRAMES
  NORMAL

DeinterlacerMode (enum)

Use Deinterlacer (DeinterlaceMode) to choose how the service will do deinterlacing. Default is Deinterlace. - Deinterlace converts interlaced to progressive. - Inverse telecine converts Hard Telecine 29.97i to progressive 23.976p. - Adaptive auto-detects and converts to progressive.

  DEINTERLACE
  INVERSE_TELECINE
  ADAPTIVE

DropFrameTimecode (enum)

Applies only to 29.97 fps outputs. When this feature is enabled, the service will use drop-frame timecode on outputs. If it is not possible to use drop-frame timecode, the system will fall back to non-drop-frame. This setting is enabled by default when Timecode insertion (TimecodeInsertion) is enabled.

  DISABLED
  ENABLED

DvbNitSettings

networkName

The network name text placed in the network_name_descriptor inside the Network Information Table. Maximum length is 256 characters.
networkId
The numeric value placed in the Network Information Table (NIT).

Type: integer
Required: True
Minimum: 0
Maximum: 65535

nitInterval
The number of milliseconds between instances of this table in the output transport stream.

Type: integer
Required: True
Minimum: 25
Maximum: 10000

DvbSdtSettings

sdtInterval
The number of milliseconds between instances of this table in the output transport stream.

Type: integer
Required: False
Minimum: 25
Maximum: 2000

serviceName
The service name placed in the service_descriptor in the Service Description Table. Maximum length is 256 characters.

Type: string
Required: False

serviceProviderName
The service provider name placed in the service_descriptor in the Service Description Table. Maximum length is 256 characters.

Type: string
Required: False

outputSdt

Type: string
Required: False
DvbSubDestinationSettings

xPosition

Specifies the horizontal position of the caption relative to the left side of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the left of the output. If no explicit x_position is provided, the horizontal caption position will be determined by the alignment parameter. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 2147483647

backgroundColor

- **Type**: string
- **Required**: False

teletextSpacing

- **Type**: string
- **Required**: False

yPosition

Specifies the vertical position of the caption relative to the top of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the top of the output. If no explicit y_position is provided, the caption will be positioned towards the bottom of the output. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 2147483647

backgroundOpacity

Specifies the opacity of the background rectangle. 255 is opaque; 0 is transparent. Leaving this parameter blank is equivalent to setting it to 0 (transparent). All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 255

fontOpacity

Specifies the opacity of the burned-in captions. 255 is opaque; 0 is transparent. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
fontResolution

Font resolution in DPI (dots per inch); default is 96 dpi. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: False
- **Minimum**: 96
- **Maximum**: 600

shadowOpacity

Specifies the opacity of the shadow. 255 is opaque; 0 is transparent. Leaving this parameter blank is equivalent to setting it to 0 (transparent). All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 255

shadowYOffset

Specifies the vertical offset of the shadow relative to the captions in pixels. A value of -2 would result in a shadow offset 2 pixels above the text. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: False
- **Minimum**: -2147483648
- **Maximum**: 2147483647

outlineSize

Specifies font outline size in pixels. This option is not valid for source captions that are either 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: True
- **Minimum**: 0
- **Maximum**: 10

outlineColor

- **Type**: string
- **Required**: True

fontSize

A positive integer indicates the exact font size in points. Set to 0 for automatic font size selection. All burn-in and DVB-Sub font settings must match.
Properties

shadowXOffset

Specifies the horizontal offset of the shadow relative to the captions in pixels. A value of -2 would result in a shadow offset 2 pixels to the left. All burn-in and DVB-Sub font settings must match.

Type: integer
Required: False
Minimum: 0
Maximum: 96

alignment

Type: string
Required: True

shadowColor

Type: string
Required: False

fontColor

Type: string
Required: False

DvbSubSourceSettings

pid

When using DVB-Sub with Burn-In or SMPTE-TT, use this PID for the source content. Unused for DVB-Sub passthrough. All DVB-Sub content is passed through, regardless of selectors.

Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

DvbSubtitleAlignment (enum)

If no explicit x_position or y_position is provided, setting alignment to centered will place the captions at the bottom center of the output. Similarly, setting a left alignment will align captions to the bottom left of the output. If x and y positions are given in conjunction with the alignment parameter, the font will be justified (either left or centered) relative to those coordinates. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

CENTERED
LEFT

**DvbSubtitleBackgroundColor (enum)**

Specifies the color of the rectangle behind the captions. All burn-in and DVB-Sub font settings must match.

NONE
BLACK
WHITE

**DvbSubtitleFontColor (enum)**

Specifies the color of the burned-in captions. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

WHITE
BLACK
YELLOW
RED
GREEN
BLUE

**DvbSubtitleOutlineColor (enum)**

Specifies font outline color. This option is not valid for source captions that are either 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

BLACK
WHITE
YELLOW
RED
GREEN
BLUE

**DvbSubtitleShadowColor (enum)**

Specifies the color of the shadow cast by the captions. All burn-in and DVB-Sub font settings must match.

NONE
BLACK
WHITE

**DvbSubtitleTeletextSpacing (enum)**

Only applies to jobs with input captions in Teletext or STL formats. Specify whether the spacing between letters in your captions is set by the captions grid or varies depending on letter width. Choose fixed grid to conform to the spacing specified in the captions file more accurately. Choose proportional to make the text easier to read if the captions are closed caption.

FIXED_GRID
PROPORTIONAL

DvbTdtSettings

tdtInterval

The number of milliseconds between instances of this table in the output transport stream.

Type: integer
Required: True
Minimum: 1000
Maximum: 30000

Eac3AttenuationControl (enum)

If set to ATTENUATE_3_DB, applies a 3 dB attenuation to the surround channels. Only used for 3/2 coding mode.

ATTENUATE_3_DB
NONE

Eac3BitstreamMode (enum)

Specifies the "Bitstream Mode" (bsmod) for the emitted E-AC-3 stream. See ATSC A/52-2012 (Annex E) for background on these values.

COMPLETE_MAIN
COMMENTARY
EMERGENCY
HEARING_IMPAIRED
VISUALLY_IMPAIRED

Eac3CodingMode (enum)

Dolby Digital Plus coding mode. Determines number of channels.

CODING_MODE_1_0
CODING_MODE_2_0
CODING_MODE_3_2

Eac3DcFilter (enum)

Activates a DC highpass filter for all input channels.

ENABLED
DISABLED

Eac3DynamicRangeCompressionLine (enum)

Enables Dynamic Range Compression that restricts the absolute peak level for a signal.

NONE
FILM_STANDARD
FILM_LIGHT
MUSIC_STANDARD
MUSIC_LIGHT
SPEECH

**Eac3DynamicRangeCompressionRf (enum)**

Enables Heavy Dynamic Range Compression, ensures that the instantaneous signal peaks do not exceed specified levels.

- NONE
- FILM_STANDARD
- FILM_LIGHT
- MUSIC_STANDARD
- MUSIC_LIGHT
- SPEECH

**Eac3LfeControl (enum)**

When encoding 3/2 audio, controls whether the LFE channel is enabled

- LFE
- NO_LFE

**Eac3LfeFilter (enum)**

Applies a 120Hz lowpass filter to the LFE channel prior to encoding. Only valid with 3_2_LFE coding mode.

- ENABLED
- DISABLED

**Eac3MetadataControl (enum)**

When set to FOLLOW_INPUT, encoder metadata will be sourced from the DD, DD+, or DolbyE decoder that supplied this audio data. If audio was not supplied from one of these streams, then the static metadata settings will be used.

- FOLLOW_INPUT
- USE_CONFIGURED

**Eac3PassthroughControl (enum)**

When set to WHEN_POSSIBLE, input DD+ audio will be passed through if it is present on the input. this detection is dynamic over the life of the transcode. Inputs that alternate between DD+ and non-DD+ content will have a consistent DD+ output as the system alternates between passthrough and encoding.

- WHEN_POSSIBLE
- NO_PASSTHROUGH

**Eac3PhaseControl (enum)**

Controls the amount of phase-shift applied to the surround channels. Only used for 3/2 coding mode.
**SHIFT_90_DEGREES**
**NO_SHIFT**

## Eac3Settings

### dialnorm
Sets the dialnorm for the output. If blank and input audio is Dolby Digital Plus, dialnorm will be passed through.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 31

### passthroughControl

- **Type:** string
- **Required:** False

### metadataControl

- **Type:** string
- **Required:** False

### bitrate
Average bitrate in bits/second. Valid bitrates depend on the coding mode.

- **Type:** integer
- **Required:** False
- **Minimum:** 64000
- **Maximum:** 640000

### dynamicRangeCompressionRf

- **Type:** string
- **Required:** False

### sampleRate
Sample rate in hz. Sample rate is always 48000.

- **Type:** integer
- **Required:** False
- **Minimum:** 48000
- **Maximum:** 48000

### surroundExMode

- **Type:** string
- **Required:** False
**Properties**

- **ltRtSurroundMixLevel**
  - Left total/Right total surround mix level. Only used for 3/2 coding mode. Valid values: -1.5 -3.0 -4.5 -6.0 -60
  - **Type:** number
  - **Required:** False
  - **Format:** float
  - **Minimum:** -60.0
  - **Maximum:** -1.5

- **dynamicRangeCompressionLine**
  - **Type:** string
  - **Required:** False

- **lfeControl**
  - **Type:** string
  - **Required:** False

- **codingMode**
  - **Type:** string
  - **Required:** False

- **surroundMode**
  - **Type:** string
  - **Required:** False

- **attenuationControl**
  - **Type:** string
  - **Required:** False

- **lfeFilter**
  - **Type:** string
  - **Required:** False

- **phaseControl**
  - **Type:** string
  - **Required:** False

- **ltRtCenterMixLevel**
  - Left total/Right total center mix level. Only used for 3/2 coding mode. Valid values: 3.0, 1.5, 0.0, -1.5 -3.0 -4.5 -6.0 -60
  - **Type:** number
  - **Required:** False
**Format**: float
**Minimum**: -60.0
**Maximum**: 3.0

**dcFilter**
- **Type**: string
- **Required**: False

**bitstreamMode**
- **Type**: string
- **Required**: False

**stereoDownmix**
- **Type**: string
- **Required**: False

**loRoSurroundMixLevel**
Left only/Right only surround mix level. Only used for 3/2 coding mode. Valid values: -1.5 -3.0 -4.5 -6.0 -60
- **Type**: number
- **Required**: False
- **Format**: float
- **Minimum**: -60.0
- **Maximum**: -1.5

**loRoCenterMixLevel**
Left only/Right only center mix level. Only used for 3/2 coding mode. Valid values: 3.0, 1.5, 0.0, -1.5 -3.0 -4.5 -6.0 -60
- **Type**: number
- **Required**: False
- **Format**: float
- **Minimum**: -60.0
- **Maximum**: 3.0

**Eac3StereoDownmix (enum)**
Stereo downmix preference. Only used for 3/2 coding mode.
- NOT_INDICATED
- LO_RO
- LT_RT
- DPL2

**Eac3SurroundExMode (enum)**
When encoding 3/2 audio, sets whether an extra center back surround channel is matrix encoded into the left and right surround channels.
Eac3SurroundMode (enum)

When encoding 2/0 audio, sets whether Dolby Surround is matrix encoded into the two channels.

NOT_INDICATED
ENABLED
DISABLED

EmbeddedConvert608To708 (enum)

When set to UPCONVERT, 608 data is both passed through via the "608 compatibility bytes" fields of the 708 wrapper as well as translated into 708. 708 data present in the source content will be discarded.

UPCONVERT
DISABLED

EmbeddedSourceSettings

source608ChannelNumber

Specifies the 608/708 channel number within the video track from which to extract captions. Unused for passthrough.

Type: integer
Required: False
Minimum: 1
Maximum: 4

counter608To708

Type: string
Required: False

source608TrackNumber

Specifies the video track index used for extracting captions. The system only supports one input video track, so this should always be set to '1'.

Type: integer
Required: False
Minimum: 1
Maximum: 1

ExceptionBody

message

Type: string
Properties

Required: False

F4vMoovPlacement (enum)
If set to PROGRESSIVE_DOWNLOAD, the MOOV atom is relocated to the beginning of the archive as required for progressive downloading. Otherwise it is placed normally at the end.

PROGRESSIVE_DOWNLOAD
NORMAL

F4vSettings

moovPlacement

Type: string
Required: False

FileGroupSettings

destination

Use Destination (Destination) to specify the S3 output location and the output filename base. Destination accepts format identifiers. If you do not specify the base filename in the URI, the service will use the filename of the input file. If your job has multiple inputs, the service uses the filename of the first input file.

Type: string
Required: False
Pattern: ^s3:\/\/

FileStreamConvert608To708 (enum)
If set to UPCONVERT, 608 caption data is both passed through via the "608 compatibility bytes" fields of the 708 wrapper as well as translated into 708. 708 data present in the source content will be discarded.

UPCONVERT
DISABLED

FileStreamSourceSettings

timeDelta

Specifies a time delta in seconds to offset the captions from the source file.

Type: integer
Required: False
Minimum: -2147483648
Maximum: 2147483647

convert608To708

Type: string
Required: False
sourceFile
External caption file used for loading captions. Accepted file extensions are 'scc', 'ttml', 'dfxp', 'stl', 'srt', and 'smi'.

   Type: string
   Required: True
   Pattern: ^(s3:\/\/(.*?\.\.(scc|SCC|ttml|TTML|dfxp|DFXP|stl|STL|srt|SRT|smi|SMI)$

FrameCaptureSettings

framerateDenominator
Frame capture will encode the first frame of the output stream, then one frame every framerateDenominator/framerateNumerator seconds. For example, settings of framerateNumerator = 1 and framerateDenominator = 3 (a rate of 1/3 frame per second) will capture the first frame, then 1 frame every 3s. Files will be named as filename.n.jpg where n is the 0-based sequence number of each Capture.

   Type: integer
   Required: False
   Minimum: 1
   Maximum: 2147483647

maxCaptures
Maximum number of captures (encoded jpg output files).

   Type: integer
   Required: False
   Minimum: 1
   Maximum: 10000000

framerateNumerator
Frame capture will encode the first frame of the output stream, then one frame every framerateDenominator/framerateNumerator seconds. For example, settings of framerateNumerator = 1 and framerateDenominator = 3 (a rate of 1/3 frame per second) will capture the first frame, then 1 frame every 3s. Files will be named as filename.NNNNNNN.jpg where N is the 0-based frame sequence number zero padded to 7 decimal places.

   Type: integer
   Required: False
   Minimum: 1
   Maximum: 2147483647

quality
JPEG Quality - a higher value equals higher quality.

   Type: integer
   Required: False
   Minimum: 1
   Maximum: 100
H264AdaptiveQuantization (enum)

Adaptive quantization. Allows intra-frame quantizers to vary to improve visual quality.

- OFF
- LOW
- MEDIUM
- HIGH
- HIGHER
- MAX

H264CodecLevel (enum)

H.264 Level.

- AUTO
- LEVEL_1
- LEVEL_1_1
- LEVEL_1_2
- LEVEL_1_3
- LEVEL_2
- LEVEL_2_1
- LEVEL_2_2
- LEVEL_3
- LEVEL_3_1
- LEVEL_3_2
- LEVEL_4
- LEVEL_4_1
- LEVEL_4_2
- LEVEL_5
- LEVEL_5_1
- LEVEL_5_2

H264CodecProfile (enum)

H.264 Profile. High 4:2:2 and 10-bit profiles are only available with the AVC-I License.

- BASELINE
- HIGH
- HIGH_10BIT
- HIGH_422
- HIGH_422_10BIT
- MAIN

H264EntropyEncoding (enum)

Entropy encoding mode. Use CABAC (must be in Main or High profile) or CAVLC.

- CABAC
- CAVLC

H264FieldEncoding (enum)

Choosing FORCE_FIELD disables PAFF encoding for interlaced outputs.
PAFF
FORCE_FIELD

**H264FlickerAdaptiveQuantization (enum)**

Adjust quantization within each frame to reduce flicker or 'pop' on I-frames.

- DISABLED
- ENABLED

**H264FramerateControl (enum)**

If you are using the console, use the Framerate setting to specify the framerate for this output. If you want to keep the same framerate as the input video, choose Follow source. If you want to do framerate conversion, choose a framerate from the dropdown list or choose Custom. The framerates shown in the dropdown list are decimal approximations of fractions. If you choose Custom, specify your framerate as a fraction. If you are creating your transcoding job specification as a JSON file without the console, use FramerateControl to specify which value the service uses for the framerate for this output. Choose INITIALIZE_FROM_SOURCE if you want the service to use the framerate from the input. Choose SPECIFIED if you want the service to use the framerate you specify in the settings FramerateNumerator and FramerateDenominator.

- INITIALIZE_FROM_SOURCE
- SPECIFIED

**H264FramerateConversionAlgorithm (enum)**

When set to INTERPOLATE, produces smoother motion during framerate conversion.

- DUPLICATE_DROP
- INTERPOLATE

**H264GopBReference (enum)**

If enable, use reference B frames for GOP structures that have B frames > 1.

- DISABLED
- ENABLED

**H264GopSizeUnits (enum)**

Indicates if the GOP Size in H264 is specified in frames or seconds. If seconds the system will convert the GOP Size into a frame count at run time.

- FRAMES
- SECONDS

**H264InterlaceMode (enum)**

Use Interlace mode (InterlaceMode) to choose the scan line type for the output. * Top Field First (TOP_FIELD) and Bottom Field First (BOTTOM_FIELD) produce interlaced output with the entire output having the same field polarity (top or bottom first). * Follow, Default Top (FOLLOW_TOP_FIELD) and Follow, Default Bottom (FOLLOW_BOTTOM_FIELD) use the same field polarity as the source. Therefore, behavior depends on the input scan type, as follows. - If the source is interlaced, the output will be interlaced with the same polarity as the source (it will follow the source). The output could therefore be a...
A mix of "top field first" and "bottom field first". If the source is progressive, the output will be interlaced with "top field first" or "bottom field first" polarity, depending on which of the Follow options you chose.

PROGRESSIVE
TOP_FIELD
BOTTOM_FIELD
FOLLOW_TOP_FIELD
FOLLOW_BOTTOM_FIELD

**H264ParControl (enum)**

Using the API, enable ParFollowSource if you want the service to use the pixel aspect ratio from the input. Using the console, do this by choosing Follow source for Pixel aspect ratio.

INITIALIZE_FROM_SOURCE
SPECIFIED

**H264QualityTuningLevel (enum)**

Use Quality tuning level (H264QualityTuningLevel) to specify whether to use fast single-pass, high-quality singlepass, or high-quality multipass video encoding.

SINGLE_PASS
SINGLE_PASS_HQ
MULTI_PASS_HQ

**H264RateControlMode (enum)**

Use this setting to specify whether this output has a variable bitrate (VBR) or constant bitrate (CBR).

VBR
CBR

**H264RepeatPps (enum)**

Places a PPS header on each encoded picture, even if repeated.

DISABLED
ENABLED

**H264SceneChangeDetect (enum)**

Scene change detection (inserts I-frames on scene changes).

DISABLED
ENABLED

**H264Settings**

slices

Number of slices per picture. Must be less than or equal to the number of macroblock rows for progressive pictures, and less than or equal to half the number of macroblock rows for interlaced pictures.
Properties

**minIInterval**

Enforces separation between repeated (cadence) I-frames and I-frames inserted by Scene Change Detection. If a scene change I-frame is within I-interval frames of a cadence I-frame, the GOP is shrunk and/or stretched to the scene change I-frame. GOP stretch requires enabling lookahead as well as setting I-interval. The normal cadence resumes for the next GOP. This setting is only used when Scene Change Detect is enabled. Note: Maximum GOP stretch = GOP size + Min-I-interval - 1

**parNumerator**

Pixel Aspect Ratio numerator.

**flickerAdaptiveQuantization**

**gopSizeUnits**

**hrdBufferSize**

Size of buffer (HRD buffer model) in bits. For example, enter five megabits as 5000000.

**qualityTuningLevel**

**maxBitrate**

Maximum bitrate in bits/second. For example, enter five megabits per second as 5000000.
Type: integer  
Required: False  
Minimum: 1000  
Maximum: 1152000000  

**bitrate**

Average bitrate in bits/second. Required for VBR and CBR. For MS Smooth outputs, bitrates must be unique when rounded down to the nearest multiple of 1000.

Type: integer  
Required: False  
Minimum: 1000  
Maximum: 1152000000  

**spatialAdaptiveQuantization**

Type: string  
Required: False  

**slowPal**

Type: string  
Required: False  

**codecProfile**

Type: string  
Required: False  

**unregisteredSeiTimecode**

Type: string  
Required: False  

**softness**

Softness. Selects quantizer matrix, larger values reduce high-frequency content in the encoded image.

Type: integer  
Required: False  
Minimum: 0  
Maximum: 128  

**framerateControl**

Type: string  
Required: False  

**telecine**

Type: string  
Required: False
codecLevel

Type: string
Required: False

framerateConversionAlgorithm

Type: string
Required: False

numberReferenceFrames

Number of reference frames to use. The encoder may use more than requested if using B-frames and/or interlaced encoding.

Type: integer
Required: False
Minimum: 1
Maximum: 6

temporalAdaptiveQuantization

Type: string
Required: False

repeatPps

Type: string
Required: False

hrdBufferInitialFillPercentage

Percentage of the buffer that should initially be filled (HRD buffer model).

Type: integer
Required: False
Minimum: 0
Maximum: 100

framerateNumerator

Framerate numerator - framerate is a fraction, e.g. 24000 / 1001 = 23.976 fps.

Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

gopClosedCadence

Frequency of closed GOPs. In streaming applications, it is recommended that this be set to 1 so a decoder joining mid-stream will receive an IDR frame as quickly as possible. Setting this value to 0 will break output segmenting.
Properties

**Type**: integer  
**Required**: False  
**Minimum**: 0  
**Maximum**: 2147483647

**numberBFramesBetweenReferenceFrames**

Number of B-frames between reference frames.

**Type**: integer  
**Required**: False  
**Minimum**: 0  
**Maximum**: 7

**framerateDenominator**

When you use the API for transcode jobs that use framerate conversion, specify the framerate as a fraction. For example, 24000 / 1001 = 23.976 fps. Use FramerateDenominator to specify the denominator of this fraction. In this example, use 1001 for the value of FramerateDenominator. When you use the console for transcode jobs that use framerate conversion, provide the value as a decimal number for Framerate. In this example, specify 23.976.

**Type**: integer  
**Required**: False  
**Minimum**: 1  
**Maximum**: 2147483647

**entropyEncoding**

**Type**: string  
**Required**: False

**fieldEncoding**

**Type**: string  
**Required**: False

**adaptiveQuantization**

**Type**: string  
**Required**: False

**interlaceMode**

**Type**: string  
**Required**: False

**gopSize**

GOP Length (keyframe interval) in frames or seconds. Must be greater than zero.

**Type**: number
### Properties

- **gopBReference**
  - **Type**: string
  - **Required**: False

- **syntax**
  - **Type**: string
  - **Required**: False

- **parDenominator**
  - **Pixel Aspect Ratio denominator.**
  - **Type**: integer
  - **Required**: False
  - **Minimum**: 1
  - **Maximum**: 2147483647

- **sceneChangeDetect**
  - **Type**: string
  - **Required**: False

- **parControl**
  - **Type**: string
  - **Required**: False

- **rateControlMode**
  - **Type**: string
  - **Required**: False

- **H264SlowPal (enum)**
  - Enables Slow PAL rate conversion. 23.976fps and 24fps input is relabeled as 25fps, and audio is sped up correspondingly.
  - **DISABLED**
  - **ENABLED**

- **H264SpatialAdaptiveQuantization (enum)**
  - Adjust quantization within each frame based on spatial variation of content complexity.
  - **DISABLED**
  - **ENABLED**
H264Syntax (enum)

Produces a bitstream compliant with SMPTE RP-2027.

DEFAULT
RP2027

H264Telecine (enum)

This field applies only if the Streams > Advanced > Framerate (framerate) field is set to 29.970. This field works with the Streams > Advanced > Preprocessors > Deinterlacer field (deinterlace_mode) and the Streams > Advanced > Interlaced Mode field (interlace_mode) to identify the scan type for the output: Progressive, Interlaced, Hard Telecine or Soft Telecine. - Hard: produces 29.97i output from 23.976 input. - Soft: produces 23.976; the player converts this output to 29.97i.

NONE
SOFT
HARD

H264TemporalAdaptiveQuantization (enum)

Adjust quantization within each frame based on temporal variation of content complexity.

DISABLED
ENABLED

H264UnregisteredSeiTimecode (enum)

Inserts timecode for each frame as 4 bytes of an unregistered SEI message.

DISABLED
ENABLED

H265AdaptiveQuantization (enum)

Adaptive quantization. Allows intra-frame quantizers to vary to improve visual quality.

OFF
LOW
MEDIUM
HIGH
HIGHER
MAX

H265AlternateTransferFunctionSei (enum)

Enables Alternate Transfer Function SEI message for outputs using Hybrid Log Gamma (HLG) Electro-Optical Transfer Function (EOTF).

DISABLED
ENABLED

H265CodecLevel (enum)

H.265 Level.
H265CodecProfile (enum)

Represents the Profile and Tier, per the HEVC (H.265) specification. Selections are grouped as [Profile] / [Tier], so "Main/High" represents Main Profile with High Tier. 4:2:2 profiles are only available with the HEVC 4:2:2 License.

- MAIN_MAIN
- MAIN_HIGH
- MAIN10_MAIN
- MAIN10_HIGH
- MAIN_422_8BIT_MAIN
- MAIN_422_8BIT_HIGH
- MAIN_422_10BIT_MAIN
- MAIN_422_10BIT_HIGH

H265FlickerAdaptiveQuantization (enum)

Adjust quantization within each frame to reduce flicker or 'pop' on I-frames.

- DISABLED
- ENABLED

H265FramerateControl (enum)

If you are using the console, use the Framerate setting to specify the framerate for this output. If you want to keep the same framerate as the input video, choose Follow source. If you want to do framerate conversion, choose a framerate from the dropdown list or choose Custom. The framerates shown in the dropdown list are decimal approximations of fractions. If you choose Custom, specify your framerate as a fraction. If you are creating your transcoding job specification as a JSON file without the console, use FramerateControl to specify which value the service uses for the framerate for this output. Choose INITIALIZE_FROM_SOURCE if you want the service to use the framerate from the input. Choose SPECIFIED if you want the service to use the framerate you specify in the settings FramerateNumerator and FramerateDenominator.

- INITIALIZE_FROM_SOURCE
- SPECIFIED

H265FramerateConversionAlgorithm (enum)

When set to INTERPOLATE, produces smoother motion during framerate conversion.
DUPLICATE_DROP
INTERPOLATE

**H265GopBReference (enum)**

If enable, use reference B frames for GOP structures that have B frames > 1.

DISABLED
ENABLED

**H265GopSizeUnits (enum)**

Indicates if the GOP Size in H265 is specified in frames or seconds. If seconds the system will convert the GOP Size into a frame count at run time.

FRAMES
SECONDS

**H265InterlaceMode (enum)**

Use Interlace mode (InterlaceMode) to choose the scan line type for the output. * Top Field First (TOP_FIELD) and Bottom Field First (BOTTOM_FIELD) produce interlaced output with the entire output having the same field polarity (top or bottom first). * Follow, Default Top (FOLLOW_TOP_FIELD) and Follow, Default Bottom (FOLLOW_BOTTOM_FIELD) use the same field polarity as the source. Therefore, behavior depends on the input scan type. - If the source is interlaced, the output will be interlaced with the same polarity as the source (it will follow the source). The output could therefore be a mix of “top field first” and “bottom field first”. - If the source is progressive, the output will be interlaced with “top field first” or “bottom field first” polarity, depending on which of the Follow options you chose.

PROGRESSIVE
TOP_FIELD
BOTTOM_FIELD
FOLLOW_TOP_FIELD
FOLLOW_BOTTOM_FIELD

**H265ParControl (enum)**

Using the API, enable ParFollowSource if you want the service to use the pixel aspect ratio from the input. Using the console, do this by choosing Follow source for Pixel aspect ratio.

INITIALIZE_FROM_SOURCE
SPECIFIED

**H265QualityTuningLevel (enum)**

Use Quality tuning level (H265QualityTuningLevel) to specify whether to use fast single-pass, high-quality singlepass, or high-quality multipass video encoding.

SINGLE_PASS
SINGLE_PASS_HQ
MULTI_PASS_HQ

**H265RateControlMode (enum)**

Use this setting to specify whether this output has a variable bitrate (VBR) or constant bitrate (CBR).
VBR
CBR

**H265SampleAdaptiveOffsetFilterMode (enum)**

Specify Sample Adaptive Offset (SAO) filter strength. Adaptive mode dynamically selects best strength based on content

- DEFAULT
- ADAPTIVE
- OFF

**H265SceneChangeDetect (enum)**

Scene change detection (inserts I-frames on scene changes).

- DISABLED
- ENABLED

**H265Settings**

**slices**

Number of slices per picture. Must be less than or equal to the number of macroblock rows for progressive pictures, and less than or equal to half the number of macroblock rows for interlaced pictures.

- Type: integer
- Required: False
- Minimum: 1
- Maximum: 32

**minIInterval**

Enforces separation between repeated (cadence) I-frames and I-frames inserted by Scene Change Detection. If a scene change I-frame is within I-interval frames of a cadence I-frame, the GOP is shrunk and/or stretched to the scene change I-frame. GOP stretch requires enabling lookahead as well as setting I-interval. The normal cadence resumes for the next GOP. This setting is only used when Scene Change Detect is enabled. Note: Maximum GOP stretch = GOP size + Min-I-interval - 1

- Type: integer
- Required: False
- Minimum: 0
- Maximum: 30

**parNumerator**

Pixel Aspect Ratio numerator.

- Type: integer
- Required: False
- Minimum: 1
Properties

**Maximum**: 2147483647

**flickerAdaptiveQuantization**

- **Type**: string
- **Required**: False

**gopSizeUnits**

- **Type**: string
- **Required**: False

**hrdBufferSize**

Size of buffer (HRD buffer model) in bits. For example, enter five megabits as 5000000.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 1466400000

**qualityTuningLevel**

- **Type**: string
- **Required**: False

**maxBitrate**

Maximum bitrate in bits/second.

- **Type**: integer
- **Required**: False
- **Minimum**: 1000
- **Maximum**: 1466400000

**bitrate**

Average bitrate in bits/second. Required for VBR and CBR. For MS Smooth outputs, bitrates must be unique when rounded down to the nearest multiple of 1000.

- **Type**: integer
- **Required**: False
- **Minimum**: 1000
- **Maximum**: 1466400000

**spatialAdaptiveQuantization**

- **Type**: string
- **Required**: False

**sampleAdaptiveOffsetFilterMode**

- **Type**: string
temporalIds
Type: string
Required: False

slowPal
Type: string
Required: False

tiles
Type: string
Required: False

writeMp4PackagingType
Type: string
Required: False

codecProfile
Type: string
Required: False

alternateTransferFunctionSei
Type: string
Required: False

unregisteredSeiTimecode
Type: string
Required: False

framerateControl
Type: string
Required: False

telecine
Type: string
Required: False

codecLevel
Type: string
Required: False
framerateConversionAlgorithm

Type: string
Required: False

numberReferenceFrames

Number of reference frames to use. The encoder may use more than requested if using B-frames and/or interlaced encoding.

Type: integer
Required: False
Minimum: 1
Maximum: 6

temporalAdaptiveQuantization

Type: string
Required: False

hrdBufferInitialFillPercentage

Percentage of the buffer that should initially be filled (HRD buffer model).

Type: integer
Required: False
Minimum: 0
Maximum: 100

framerateNumerator

Framerate numerator - framerate is a fraction, e.g. 24000 / 1001 = 23.976 fps.

Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

gopClosedCadence

Frequency of closed GOPs. In streaming applications, it is recommended that this be set to 1 so a decoder joining mid-stream will receive an IDR frame as quickly as possible. Setting this value to 0 will break output segmenting.

Type: integer
Required: False
Minimum: 0
Maximum: 2147483647

numberBFramesBetweenReferenceFrames

Number of B-frames between reference frames.
Type: integer  
Required: False  
Minimum: 0  
Maximum: 7

framerateDenominator

Framerate denominator.

Type: integer  
Required: False  
Minimum: 1  
Maximum: 2147483647

adaptiveQuantization

Type: string  
Required: False

interlaceMode

Type: string  
Required: False

gopSize

GOP Length (keyframe interval) in frames or seconds. Must be greater than zero.

Type: number  
Required: False  
Format: float  
Minimum: 0.0

gopBReference

Type: string  
Required: False

parDenominator

Pixel Aspect Ratio denominator.

Type: integer  
Required: False  
Minimum: 1  
Maximum: 2147483647

sceneChangeDetect

Type: string  
Required: False
parControl
- Type: string
- Required: False

rateControlMode
- Type: string
- Required: False

H265SlowPal (enum)
Enables Slow PAL rate conversion. 23.976fps and 24fps input is relabeled as 25fps, and audio is sped up correspondingly.
- DISABLED
- ENABLED

H265SpatialAdaptiveQuantization (enum)
Adjust quantization within each frame based on spatial variation of content complexity.
- DISABLED
- ENABLED

H265Telecine (enum)
This field applies only if the Streams > Advanced > Framerate (framerate) field is set to 29.970. This field works with the Streams > Advanced > Preprocessors > Deinterlacer field (deinterlace_mode) and the Streams > Advanced > Interlaced Mode field (interlace_mode) to identify the scan type for the output:
- Progressive, Interlaced, Hard Telecine or Soft Telecine.
- Hard: produces 29.97i output from 23.976 input.
- Soft: produces 23.976; the player converts this output to 29.97i.
- NONE
- SOFT
- HARD

H265TemporalAdaptiveQuantization (enum)
Adjust quantization within each frame based on temporal variation of content complexity.
- DISABLED
- ENABLED

H265TemporalIds (enum)
Enables temporal layer identifiers in the encoded bitstream. Up to 3 layers are supported depending on GOP structure: I- and P-frames form one layer, reference B-frames can form a second layer and non-reference b-frames can form a third layer. Decoders can optionally decode only the lower temporal layers to generate a lower frame rate output. For example, given a bitstream with temporal IDs and with b-frames = 1 (i.e. IbpPb display order), a decoder could decode all the frames for full frame rate output or only the I and P frames (lowest temporal layer) for a half frame rate output.
- DISABLED
- ENABLED
H265Tiles (enum)
Enable use of tiles, allowing horizontal as well as vertical subdivision of the encoded pictures.
- DISABLED
- ENABLED

H265UnregisteredSeiTimecode (enum)
Inserts timecode for each frame as 4 bytes of an unregistered SEI message.
- DISABLED
- ENABLED

H265WriteMp4PackagingType (enum)
If HVC1, output that is H.265 will be marked as HVC1 and adhere to the ISO-IEC/TC1-SC29_N13798_Text_ISOIEC_FDIS_14496-15_3rd_E spec which states that parameter set NAL units will be stored in the sample headers but not in the samples directly. If HEV1, then H.265 will be marked as HEV1 and parameter set NAL units will be written into the samples.
- HVC1
- HEV1

Hdr10Metadata

redPrimaryY
HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

Type: integer
Required: False
Minimum: 0
Maximum: 50000

greenPrimaryY
HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

Type: integer
Required: False
Minimum: 0
Maximum: 50000

whitePointX
HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

Type: integer
Properties

maxLuminance
Nominal maximum mastering display luminance in units of 0.0001 candelas per square meter.

Type: integer
Required: False
Minimum: 0
Maximum: 50000

greenPrimaryX
HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

Type: integer
Required: False
Minimum: 0
Maximum: 50000

whitePointY
HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

Type: integer
Required: False
Minimum: 0
Maximum: 50000

redPrimaryX
HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

Type: integer
Required: False
Minimum: 0
Maximum: 50000

bluePrimaryX
HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

Type: integer
Required: False
Minimum: 0
Maximum: 50000

bluePrimaryY

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

Type: integer
Required: False
Minimum: 0
Maximum: 50000

maxFrameAverageLightLevel

Maximum average light level of any frame in the coded video sequence, in units of candelas per square meter.

Type: integer
Required: True
Minimum: 0
Maximum: 65535

maxContentLightLevel

Maximum light level among all samples in the coded video sequence, in units of candelas per square meter.

Type: integer
Required: True
Minimum: 0
Maximum: 65535

minLuminance

Nominal minimum mastering display luminance in units of 0.0001 candelas per square meter.

Type: integer
Required: False
Minimum: 0
Maximum: 2147483647

HlsAdMarkers (enum)

ELEMENTAL
ELEMENTAL_SCTE35

HlsAudioTrackType (enum)

Four types of audio-only tracks are supported: Audio-Only Variant Stream The client can play back this audio-only stream instead of video in low-bandwidth scenarios. Represented as an EXT-X-STREAM-INF in the HLS manifest. Alternate Audio, Auto Select, Default Alternate rendition that the client should try to play back by default. Represented as an EXT-X-MEDIA in the HLS manifest with DEFAULT=YES, AUTOSELECT=YES Alternate Audio, Auto Select, Not Default Alternate rendition that the client may
try to play back by default. Represented as an EXT-X-MEDIA in the HLS manifest with DEFAULT=NO, AUTOSELECT=YES Alternate Audio, not Auto Select Alternate rendition that the client will not try to play back by default. Represented as an EXT-X-MEDIA in the HLS manifest with DEFAULT=NO, AUTOSELECT=NO

ALTERNATE_AUDIO_AUTO_SELECT_DEFAULT
ALTERNATE_AUDIO_AUTO_SELECT
ALTERNATE_AUDIO_NOT_AUTO_SELECT
AUDIO_ONLY_VARIANT_STREAM

**HlsCaptionLanguageMapping**

**languageDescription**

Caption language description.

*Type:* string  
*Required:* False

**captionChannel**

Caption channel.

*Type:* integer  
*Required:* False  
*Minimum:* -2147483648  
*Maximum:* 2147483647

**languageCode**

*Type:* string  
*Required:* False

**customLanguageCode**

Specify the language for this caption channel, using the ISO 639-2 or ISO 639-3 three-letter language code.

*Type:* string  
*Required:* False  
*Pattern:* \^[A-Za-z]{3}$

**HlsCaptionLanguageSetting (enum)**

Applies only to 608 Embedded output captions. Insert: Include CLOSED-CAPTIONS lines in the manifest. Specify at least one language in the CC1 Language Code field. One CLOSED-CAPTION line is added for each Language Code you specify. Make sure to specify the languages in the order in which they appear in the original source (if the source is embedded format) or the order of the caption selectors (if the source is other than embedded). Otherwise, languages in the manifest will not match up properly with the output captions. None: Include CLOSED-CAPTIONS=None line in the manifest. Omit: Omit any CLOSED-CAPTIONS line from the manifest.

INSERT
OMIT
NONE

HlsClientCache (enum)
When set to ENABLED, sets #EXT-X-ALLOW-CACHE:no tag, which prevents client from saving media
segments for later replay.
   DISABLED
   ENABLED

HlsCodecSpecification (enum)
Specification to use (RFC-6381 or the default RFC-4281) during m3u8 playlist generation.
   RFC_6381
   RFC_4281

HlsDirectoryStructure (enum)
Indicates whether segments should be placed in subdirectories.
   SINGLE_DIRECTORY
   SUBDIRECTORY_PER_STREAM

HlsEncryptionSettings
initializationVectorInManifest
   Type: string
   Required: False
constantInitializationVector
This is a 128-bit, 16-byte hex value represented by a 32-character text string. If this parameter is not set
then the Initialization Vector will follow the segment number by default.
   Type: string
   Required: False
   Pattern: ^[0-9a-fA-F]{32}$
staticKeyProvider
   Type: StaticKeyProvider (p. 189)
   Required: False
type
   Type: string
   Required: True
encryptionMethod
   Type: string
Properties

**Required**: False

**spekeKeyProvider**

*Type*: SpekeKeyProvider (p. 189)

*Required*: False

**HlsEncryptionType (enum)**

Encrypts the segments with the given encryption scheme. Leave blank to disable. Selecting ‘Disabled’ in the web interface also disables encryption.

- AES128
- SAMPLE_AES

**HlsGroupSettings**

**segmentsPerSubdirectory**

Number of segments to write to a subdirectory before starting a new one. directoryStructure must be SINGLE_DIRECTORY for this setting to have an effect.

*Type*: integer

*Required*: False

*Minimum*: 1

*Maximum*: 2147483647

**streamInfResolution**

*Type*: string

*Required*: False

**timestampDeltaMilliseconds**

Provides an extra millisecond delta offset to fine tune the timestamps.

*Type*: integer

*Required*: False

*Minimum*: -2147483648

*Maximum*: 2147483647

**outputSelection**

*Type*: string

*Required*: False

**captionLanguageMappings**

Language to be used on Caption outputs

*Type*: Array of type HlsCaptionLanguageMapping (p. 139)

*Required*: False
clientCache
  Type: string
  Required: False

codecSpecification
  Type: string
  Required: False

destination
Use Destination (Destination) to specify the S3 output location and the output filename base. Destination accepts format identifiers. If you do not specify the base filename in the URI, the service will use the filename of the input file. If your job has multiple inputs, the service uses the filename of the first input file.
  Type: string
  Required: False
  Pattern: ^s3:\/\/

timedMetadataId3Frame
  Type: string
  Required: False

segmentControl
  Type: string
  Required: False

timedMetadataId3Period
Timed Metadata interval in seconds.
  Type: integer
  Required: False
  Minimum: -2147483648
  Maximum: 2147483647

captionLanguageSetting
  Type: string
  Required: False

minSegmentLength
When set, Minimum Segment Size is enforced by looking ahead and back within the specified range for a nearby avail and extending the segment size if needed.
  Type: integer
  Required: True
  Minimum: 0
Maximum: 2147483647

directoryStructure

Type: string
Required: False

programDateTime

Type: string
Required: False

baseUrl

A partial URI prefix that will be prepended to each output in the media .m3u8 file. Can be used if base manifest is delivered from a different URL than the main .m3u8 file.

Type: string
Required: False

encryption

DRM settings.

Type: HlsEncryptionSettings (p. 140)
Required: False

adMarkers

Choose one or more ad marker types to pass SCTE35 signals through to this group of Apple HLS outputs.

Type: Array of type string
Required: False

programDateTimePeriod

Period of insertion of EXT-X-PROGRAM-DATE-TIME entry, in seconds.

Type: integer
Required: False
Minimum: 0
Maximum: 3600

manifestCompression

Type: string
Required: False

segmentLength

Length of MPEG-2 Transport Stream segments to create (in seconds). Note that segments will end on the next keyframe after this number of seconds, so actual segment length may be longer.
Type: integer
Required: True
Minimum: 1
Maximum: 2147483647

manifestDurationFormat
Type: string
Required: False

HlsIFrameOnlyManifest (enum)
When set to INCLUDE, writes I-Frame Only Manifest in addition to the HLS manifest
- INCLUDE
- EXCLUDE

HlsInitializationVectorInManifest (enum)
The Initialization Vector is a 128-bit number used in conjunction with the key for encrypting blocks. If set to INCLUDE, Initialization Vector is listed in the manifest. Otherwise Initialization Vector is not in the manifest.
- INCLUDE
- EXCLUDE

HlsKeyProviderType (enum)
Indicates which type of key provider is used for encryption.
- SPEKE
- STATIC_KEY

HlsManifestCompression (enum)
When set to GZIP, compresses HLS playlist.
- GZIP
- NONE

HlsManifestDurationFormat (enum)
Indicates whether the output manifest should use floating point values for segment duration.
- FLOATING_POINT
- INTEGER

HlsOutputSelection (enum)
Indicates whether the .m3u8 manifest file should be generated for this HLS output group.
- MANIFESTS_AND_SEGMENTS
SEGMENTS_ONLY

**HlsProgramDateTime (enum)**

Includes or excludes EXT-X-PROGRAM-DATE-TIME tag in .m3u8 manifest files. The value is calculated as follows: either the program date and time are initialized using the input timecode source, or the time is initialized using the input timecode source and the date is initialized using the timestamp_offset.

- INCLUDE
- EXCLUDE

**HlsSegmentControl (enum)**

When set to SINGLE_FILE, emits program as a single media resource (.ts) file, uses #EXT-X-BYTERANGE tags to index segment for playback.

- SINGLE_FILE
- SEGMENTED_FILES

**HlsSettings**

**iFrameOnlyManifest**

- **Type**: string
- **Required**: False

**audioRenditionSets**

List all the audio groups that are used with the video output stream. Input all the audio GROUP-IDs that are associated to the video, separate by ",".

- **Type**: string
- **Required**: False

**audioTrackType**

- **Type**: string
- **Required**: False

**segmentModifier**

String concatenated to end of segment filenames. Accepts "Format Identifiers":#format_identifier_parameters.

- **Type**: string
- **Required**: False

**audioGroupId**

Specifies the group to which the audio Rendition belongs.

- **Type**: string
- **Required**: False
**HlsStreamInfResolution (enum)**

Include or exclude RESOLUTION attribute for video in EXT-X-STREAM-INF tag of variant manifest.

- INCLUDE
- EXCLUDE

**HlsTimedMetadataId3Frame (enum)**

Indicates ID3 frame that has the timecode.

- NONE
- PRIV
- TDRL

**Id3Insertion**

**id3**

Use ID3 tag (Id3) to provide a tag value in base64-encode format.

- **Type**: string
- **Required**: True
- **Pattern**: `^[A-Za-z0-9+\/-]+={0,2}$`

**timecode**

Provide a Timecode (TimeCode) in HH:MM:SS:FF or HH:MM:SS;FF format.

- **Type**: string
- **Required**: True
- **Format**: timecode
- **Pattern**: `^[01][0-9]|2[0-4]):[0-5][0-9];[0-5][0-9][;][0-9]{2}$`

**ImageInserter**

**insertableImages**

Image to insert. Must be 32 bit windows BMP, PNG, or TGA file. Must not be larger than the output frames.

- **Type**: Array of type InsertableImage (p. 149)
- **Required**: True

**InputClipping**

**startTimecode**

Set Start timecode (StartTimecode) to the beginning of the portion of the input you are clipping. The frame corresponding to the Start timecode value is included in the clip. Start timecode or End timecode may be left blank, but not both. Use the format HH:MM:SS:FF or HH:MM:SS;FF, where HH is the hour, MM is the minute, SS is the second, and FF is the frame number. When choosing this value, take into account your setting for Input timecode source. For example, if you have embedded timecodes that start at 01:00:00:00 and you want your clip to begin five minutes into the video, use 01:05:00:00.
**Properties**

**Type**: string  
**Required**: False  
**Format**: timecode  
**Pattern**: `^([01][0-9]|2[0-4]):[0-5][0-9]:[0-5][0-9][;][0-9]{2}$`

**endTimecode**

Set End timecode (EndTimecode) to the end of the portion of the input you are clipping. The frame corresponding to the End timecode value is included in the clip. Start timecode or End timecode may be left blank, but not both. Use the format HH:MM:SS:FF or HH:MM:SS;FF, where HH is the hour, MM is the minute, SS is the second, and FF is the frame number. When choosing this value, take into account your setting for timecode source under input settings (InputTimecodeSource). For example, if you have embedded timecodes that start at 01:00:00:00 and you want your clip to end six minutes into the video, use 01:06:00:00.

**Type**: string  
**Required**: False  
**Format**: timecode  
**Pattern**: `^([01][0-9]|2[0-4]):[0-5][0-9]:[0-5][0-9][;][0-9]{2}$`

**InputDeblockFilter (enum)**

Enable Deblock (InputDeblockFilter) to produce smoother motion in the output. Default is disabled. Only manually controllable for MPEG2 and uncompressed video inputs.

- ENABLED
- DISABLED

**InputDenoiseFilter (enum)**

Enable Denoise (InputDenoiseFilter) to filter noise from the input. Default is disabled. Only applicable to MPEG2, H.264, H.265, and uncompressed video inputs.

- ENABLED
- DISABLED

**InputFilterEnable (enum)**

Use Filter enable (InputFilterEnable) to specify how the transcoding service applies the denoise and deblock filters. You must also enable the filters separately, with Denoise (InputDenoiseFilter) and Debloc (InputDeblockFilter).  
* AUTO - The transcoding service determines whether to apply filtering, depending on input type and quality.  
* DISABLE - The input is not filtered. This is true even if you use the API to enable them in (InputDeblockFilter) and (InputDeblockFilter).  
* FORCE - The input is filtered regardless of input type.

- AUTO
- DISABLE
- FORCE

**InputPsiControl (enum)**

Set PSI control (InputPsiControl) for transport stream inputs to specify which data the demux process to scans.  
* IGNORE PSI - Scan all PIDs for audio and video.  
* USE PSI - Scan only PSI data.

- IGNORE_PSI
- USE_PSI
InputTemplate

audioSelectors

Use Audio selectors (AudioSelectors) to specify a track or set of tracks from the input that you will use in your outputs. You can use multiple Audio selectors per input.

  Type: object
  Required: False

audioSelectorGroups

Specifies set of audio selectors within an input to combine. An input may have multiple audio selector groups. See "Audio Selector Group":#inputs-audio_selector_group for more information.

  Type: object
  Required: False

filterEnable

  Type: string
  Required: False

deblockFilter

  Type: string
  Required: False

videoSelector

  Type: VideoSelector (p. 197)
  Required: False

filterStrength

Use Filter strength (FilterStrength) to adjust the magnitude the input filter settings (Deblock and Denoise). The range is -5 to 5. Default is 0.

  Type: integer
  Required: False
  Minimum: -5
  Maximum: 5

programNumber

Use Program (programNumber) to select a specific program from within a multi-program transport stream. Note that Quad 4K is not currently supported. Default is the first program within the transport stream. If the program you specify doesn't exist, the transcoding service will use this default.

  Type: integer
  Required: False
  Minimum: 1
  Maximum: 2147483647
**timecodeSource**

*Type:* string  
*Required:* False

**denoiseFilter**

*Type:* string  
*Required:* False

**captionSelectors**

Use Captions selectors (CaptionSelectors) to specify the captions data from the input that you will use in your outputs. You can use multiple captions selectors per input.

*Type:* object  
*Required:* False

**inputClippings**

(InputClippings) contains sets of start and end times that together specify a portion of the input to be used in the outputs. If you provide only a start time, the clip will be the entire input from that point to the end. If you provide only an end time, it will be the entire input up to that point. When you specify more than one input clip, the transcoding service creates the job outputs by stringing the clips together in the order you specify them.

*Type:* Array of type InputClipping (p. 146)  
*Required:* False

**psiControl**

*Type:* string  
*Required:* False

**InputTimecodeSource (enum)**

Timecode source under input settings (InputTimecodeSource) only affects the behavior of features that apply to a single input at a time, such as input clipping and synchronizing some captions formats. Use this setting to specify whether the service counts frames by timecodes embedded in the video (EMBEDDED) or by starting the first frame at zero (ZEROBASED). In both cases, the timecode format is HH:MM:SS:FF or HH:MM:SS;FF, where FF is the frame number. Only set this to EMBEDDED if your source video has embedded timecodes.

- EMBEDDED
- ZEROBASED
- SPECIFIEDSTART

**InsertableImage**

**duration**

Use Duration (Duration) to set the time, in milliseconds, for the image to remain on the output video.

*Type:* integer
Properties

**Required**: False  
**Minimum**: -2147483648  
**Maximum**: 2147483647

**fadeOut**

Use Fade out (FadeOut) to set the length, in milliseconds, of the inserted image fade out. If you don’t specify a value for Fade out, the image will disappear abruptly at the end of the inserted image duration.

- **Type**: integer  
- **Required**: False  
- **Minimum**: -2147483648  
- **Maximum**: 2147483647

**imageY**

Use Top (ImageY) to set the distance, in pixels, between the inserted image and the top edge of the video frame. Required for BMP, PNG and TGA input.

- **Type**: integer  
- **Required**: True  
- **Minimum**: -2147483648  
- **Maximum**: 2147483647

**fadeIn**

Use Fade in (FadeIn) to set the length, in milliseconds, of the inserted image fade in. If you don't specify a value for Fade in, the image will appear abruptly at the Start time.

- **Type**: integer  
- **Required**: False  
- **Minimum**: -2147483648  
- **Maximum**: 2147483647

**imageX**

Use Left (ImageX) to set the distance, in pixels, between the inserted image and the left edge of the frame. Required for BMP, PNG and TGA input.

- **Type**: integer  
- **Required**: True  
- **Minimum**: -2147483648  
- **Maximum**: 2147483647

**width**

Specify the Width (Width) of the inserted image. Use a value that is less than or equal to the video resolution width. Leave this setting blank to use the native width of the image.

- **Type**: integer  
- **Required**: False  
- **Minimum**: -2147483648  
- **Maximum**: 2147483647
startTime

Use Start time (StartTime) to specify the video timecode when the image is inserted in the output. This must be in timecode (HH:MM:SS:FF or HH:MM:SS;FF) format.

  Type: string
  Required: False
  Pattern: ^(((\[0-1]\d)|(2[0-3]))((0-5)\d){2}((;[0-5]\d))$}

opacity

Use Opacity (Opacity) to specify how much of the underlying video shows through the inserted image. 0 is transparent and 100 is fully opaque. Default is 50.

  Type: integer
  Required: True
  Minimum: 0
  Maximum: 100

layer

Use Layer (Layer) to specify how overlapping inserted images appear. Images with higher values of layer appear on top of images with lower values of layer.

  Type: integer
  Required: True
  Minimum: 0
  Maximum: 99

height

Specify the Height (Height) of the inserted image. Use a value that is less than or equal to the video resolution height. Leave this setting blank to use the native height of the image.

  Type: integer
  Required: False
  Minimum: -2147483648
  Maximum: 2147483647

imageInserterInput

Use Image location (imageInserterInput) to specify the Amazon S3 location of the image to be inserted into the output. Use a 32 bit BMP, PNG, or TGA file that fits inside the video frame.

  Type: string
  Required: True
  Pattern: ^(s3:\/\/[.*])(bmp|BMP|png|PNG|tga|TGA)$

JobTemplate

createdAt

The timestamp in epoch seconds for Job template creation.
Properties

**Type**
Type: string  
Required: False  
Format: date-time

**lastUpdated**
The timestamp in epoch seconds when the Job template was last updated.

Type: string  
Required: False  
Format: date-time

**settings**
Type: JobTemplateSettings (p. 153)  
Required: True

**name**
A name you create for each job template. Each name must be unique within your account.

Type: string  
Required: True

**description**
An optional description you create for each job template.

Type: string  
Required: False

**arn**
An identifier for this resource that is unique within all of AWS.

Type: string  
Required: False

**category**
An optional category you create to organize your job templates.

Type: string  
Required: False

**type**
A job template can be of two types: system or custom. System or built-in job templates can't be modified or deleted by the user.

Type: string  
Required: False
queue

Optional. The queue that jobs created from this template are assigned to. If you don't specify this, jobs will go to the default queue.

Type: string
Required: False

JobTemplateListBy (enum)

Optional. When you request a list of job templates, you can choose to list them alphabetically by NAME or chronologically by CREATION_DATE. If you don't specify, the service will list them by name.

NAME
CREATION_DATE
SYSTEM

JobTemplateSettings

timecodeConfig

Contains settings used to acquire and adjust timecode information from inputs.

Type: TimecodeConfig (p. 191)
Required: False

adAvailOffset

When specified, this offset (in milliseconds) is added to the input Ad Avail PTS time.

Type: integer
Required: False
Minimum: -1000
Maximum: 1000

outputGroups

(OutputGroups) contains one group of settings for each set of outputs that share a common package type. All unpackaged files (MPEG-4, MPEG-2 TS, Quicktime, MXF, and no container) are grouped in a single output group as well. Required in (OutputGroups) is a group of settings that apply to the whole group. This required object depends on the value you set for (Type) under (OutputGroups)->(OutputGroupSettings). Type, settings object pairs are as follows. * FILE_GROUP_SETTINGS, FileGroupSettings * HLS_GROUP_SETTINGS, HlsGroupSettings * DASH_ISO_GROUP_SETTINGS, DashIsoGroupSettings * MS_SMOOTH_GROUP_SETTINGS, MsSmoothGroupSettings * CMAF_GROUP_SETTINGS, CmafGroupSettings

Type: Array of type OutputGroup (p. 182)
Required: True

nielsenConfiguration

Type: NielsenConfiguration (p. 179)
Required: False
inputs

Use Inputs (inputs) to define the source file used in the transcode job. There can only be one input in a job template. Using the API, you can include multiple inputs when referencing a job template.

  Type: Array of type InputTemplate (p. 148)
  Required: False

timedMetadataInsertion

  Type: TimedMetadataInsertion (p. 192)
  Required: False

availBlanking

Settings for ad avail blanking. Video can be blanked or overlaid with an image, and audio muted during SCTE-35 triggered ad avails.

  Type: AvailBlanking (p. 88)
  Required: False

LanguageCode (enum)


  ENG
  SPA
  FRA
  DEU
  GER
  ZHO
  ARA
  HIN
  JPN
  RUS
  POR
  ITA
  URD
  VIE
  KOR
  PAN
  ABK
  AAR
  AFR
  AKA
  SQI
  AMH
  ARG
  HYE
  ASM
  AVA
  AVE
  AYM
  AZE
  BAM
<table>
<thead>
<tr>
<th>Code</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAK</td>
<td></td>
</tr>
<tr>
<td>EUS</td>
<td></td>
</tr>
<tr>
<td>BEL</td>
<td></td>
</tr>
<tr>
<td>BEN</td>
<td></td>
</tr>
<tr>
<td>BENH</td>
<td></td>
</tr>
<tr>
<td>BIS</td>
<td></td>
</tr>
<tr>
<td>BOS</td>
<td></td>
</tr>
<tr>
<td>BRE</td>
<td></td>
</tr>
<tr>
<td>BUL</td>
<td></td>
</tr>
<tr>
<td>MYA</td>
<td></td>
</tr>
<tr>
<td>CAT</td>
<td></td>
</tr>
<tr>
<td>KHM</td>
<td></td>
</tr>
<tr>
<td>CHA</td>
<td></td>
</tr>
<tr>
<td>CHE</td>
<td></td>
</tr>
<tr>
<td>NYA</td>
<td></td>
</tr>
<tr>
<td>CHU</td>
<td></td>
</tr>
<tr>
<td>CHV</td>
<td></td>
</tr>
<tr>
<td>COR</td>
<td></td>
</tr>
<tr>
<td>COS</td>
<td></td>
</tr>
<tr>
<td>CRE</td>
<td></td>
</tr>
<tr>
<td>HRV</td>
<td></td>
</tr>
<tr>
<td>CES</td>
<td></td>
</tr>
<tr>
<td>DAN</td>
<td></td>
</tr>
<tr>
<td>DIV</td>
<td></td>
</tr>
<tr>
<td>NLD</td>
<td></td>
</tr>
<tr>
<td>DZO</td>
<td></td>
</tr>
<tr>
<td>ENM</td>
<td></td>
</tr>
<tr>
<td>EPO</td>
<td></td>
</tr>
<tr>
<td>EST</td>
<td></td>
</tr>
<tr>
<td>EWE</td>
<td></td>
</tr>
<tr>
<td>FAO</td>
<td></td>
</tr>
<tr>
<td>FIJ</td>
<td></td>
</tr>
<tr>
<td>FIN</td>
<td></td>
</tr>
<tr>
<td>FRM</td>
<td></td>
</tr>
<tr>
<td>FUL</td>
<td></td>
</tr>
<tr>
<td>GLA</td>
<td></td>
</tr>
<tr>
<td>GLG</td>
<td></td>
</tr>
<tr>
<td>LUG</td>
<td></td>
</tr>
<tr>
<td>LUG</td>
<td></td>
</tr>
<tr>
<td>KAT</td>
<td></td>
</tr>
<tr>
<td>ELL</td>
<td></td>
</tr>
<tr>
<td>GRN</td>
<td></td>
</tr>
<tr>
<td>GUJ</td>
<td></td>
</tr>
<tr>
<td>HAT</td>
<td></td>
</tr>
<tr>
<td>HAU</td>
<td></td>
</tr>
<tr>
<td>HEB</td>
<td></td>
</tr>
<tr>
<td>HER</td>
<td></td>
</tr>
<tr>
<td>HMO</td>
<td></td>
</tr>
<tr>
<td>HUN</td>
<td></td>
</tr>
<tr>
<td>ISL</td>
<td></td>
</tr>
<tr>
<td>IDO</td>
<td></td>
</tr>
<tr>
<td>IBO</td>
<td></td>
</tr>
<tr>
<td>IND</td>
<td></td>
</tr>
<tr>
<td>INA</td>
<td></td>
</tr>
<tr>
<td>ILE</td>
<td></td>
</tr>
<tr>
<td>IKU</td>
<td></td>
</tr>
<tr>
<td>IPK</td>
<td></td>
</tr>
</tbody>
</table>
ListJobTemplatesRequest

nextToken

Use this string, provided with the response to a previous request, to request the next batch of job templates.
Type: string
Required: False

maxResults
Optional. Number of job templates, up to twenty, that will be returned at one time.
Type: integer
Required: False
Format: int32
Minimum: 1
Maximum: 20

listBy
Type: string
Required: False

category
Optionally, specify a job template category to limit responses to only job templates from that category.
Type: string
Required: False

order
Type: string
Required: False

ListJobTemplatesResponse

nextToken
Use this string to request the next batch of job templates.
Type: string
Required: False

jobTemplates
List of Job templates.
Type: Array of type JobTemplate (p. 151)
Required: False

M2tsAudioBufferModel (enum)
Selects between the DVB and ATSC buffer models for Dolby Digital audio.
DVB
ATSC

**M2tsBufferModel (enum)**

Controls what buffer model to use for accurate interleaving. If set to MULTIPLEX, use multiplex buffer model. If set to NONE, this can lead to lower latency, but low-memory devices may not be able to playback the stream without interruptions.

MULITPLEX
NONE

**M2tsEbpAudioInterval (enum)**

When set to VIDEO_AND_FIXED_INTERVALS, audio EBP markers will be added to partitions 3 and 4. The interval between these additional markers will be fixed, and will be slightly shorter than the video EBP marker interval. When set to VIDEO_INTERVAL, these additional markers will not be inserted. Only applicable when EBP segmentation markers are selected (segmentationMarkers is EBP or EBP_LEGACY).

VIDEO_AND_FIXED_INTERVALS
VIDEO_INTERVAL

**M2tsEbpPlacement (enum)**

Selects which PIDs to place EBP markers on. They can either be placed only on the video PID, or on both the video PID and all audio PIDs. Only applicable when EBP segmentation markers are selected (segmentationMarkers is EBP or EBP_LEGACY).

VIDEO_AND_AUDIO_PIDS
VIDEO_PID

**M2tsEsRateInPES (enum)**

Controls whether to include the ES Rate field in the PES header.

INCLUDE
EXCLUDE

**M2ts NielsenId3 (enum)**

If INSERT, Nielsen inaudible tones for media tracking will be detected in the input audio and an equivalent ID3 tag will be inserted in the output.

INSERT
NONE

**M2tsPcrControl (enum)**

When set to PCR_EVERY_PES_PACKET, a Program Clock Reference value is inserted for every Packetized Elementary Stream (PES) header. This is effective only when the PCR PID is the same as the video or audio elementary stream.

PCR_EVERY_PES_PACKET
CONFIGURED_PCR_PERIOD
**M2tsRateMode** (enum)

When set to CBR, inserts null packets into transport stream to fill specified bitrate. When set to VBR, the bitrate setting acts as the maximum bitrate, but the output will not be padded up to that bitrate.

VBR
CBR

**M2tsScte35Source** (enum)

Enables SCTE-35 passthrough (scte35Source) to pass any SCTE-35 signals from input to output.

PASSTHROUGH
NONE

**M2tsSegmentationMarkers** (enum)

Inserts segmentation markers at each segmentation_time period. rai_segstart sets the Random Access Indicator bit in the adaptation field. rai_adapt sets the RAI bit and adds the current timecode in the private data bytes. psi_segstart inserts PAT and PMT tables at the start of segments. ebp adds Encoder Boundary Point information to the adaptation field as per OpenCable specification OC-SP-EBP-I01-130118. ebp_legacy adds Encoder Boundary Point information to the adaptation field using a legacy proprietary format.

NONE
RAI_SEGSTART
RAI_ADAPT
PSI_SEGSTART
EBP
EBP_LEGACY

**M2tsSegmentationStyle** (enum)

The segmentation style parameter controls how segmentation markers are inserted into the transport stream. With avails, it is possible that segments may be truncated, which can influence where future segmentation markers are inserted. When a segmentation style of "reset_cadence" is selected and a segment is truncated due to an avail, we will reset the segmentation cadence. This means the subsequent segment will have a duration of $segmentation_time seconds. When a segmentation style of "maintain_cadence" is selected and a segment is truncated due to an avail, we will not reset the segmentation cadence. This means the subsequent segment will likely be truncated as well. However, all segments after that will have a duration of $segmentation_time seconds. Note that EBP lookahead is a slight exception to this rule.

MAINTAIN_CADENCE
RESET_CADENCE

**M2tsSettings**

dvbTeletextPid

Packet Identifier (PID) for input source DVB Teletext data to this output.

**Type:** integer
**Required:** False
**Minimum:** 32
Maximum: 8182

nielsenId3
Type: string
Required: False

bitrate
The output bitrate of the transport stream in bits per second. Setting to 0 lets the muxer automatically determine the appropriate bitrate. Other common values are 3750000, 7500000, and 15000000.
Type: integer
Required: False
Minimum: 0
Maximum: 2147483647

segmentationTime
The length in seconds of each segment. Required unless markers is set to _none_.
Type: number
Required: False
Format: float
Minimum: 0.0

audioPids
Packet Identifier (PID) of the elementary audio stream(s) in the transport stream. Multiple values are accepted, and can be entered in ranges and/or by comma separation.
Type: Array of type integer
Required: False

rateMode
Type: string
Required: False

maxPcrInterval
Maximum time in milliseconds between Program Clock References (PCRs) inserted into the transport stream.
Type: integer
Required: False
Minimum: 0
Maximum: 500

audioFramesPerPes
The number of audio frames to insert for each PES packet.
Type: integer
Required: False
Minimum: 0
Maximum: 2147483647

ebpAudioInterval
Type: string
Required: False

fragmentTime
The length in seconds of each fragment. Only used with EBP markers.
Type: number
Required: False
Format: float
Minimum: 0.0

scte35Pid
Packet Identifier (PID) of the SCTE-35 stream in the transport stream.
Type: integer
Required: False
Minimum: 32
Maximum: 8182

pmtInterval
The number of milliseconds between instances of this table in the output transport stream.
Type: integer
Required: False
Minimum: 0
Maximum: 1000

privateMetadataPid
Packet Identifier (PID) of the private metadata stream in the transport stream.
Type: integer
Required: False
Minimum: 32
Maximum: 8182

segmentationStyle
Type: string
Required: False

audioBufferModel
Type: string
**Properties**

**Required**: False

**programNumber**

The value of the program number field in the Program Map Table.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 65535

**dvbNitSettings**

- **Type**: `DvbNitSettings` (p. 105)
- **Required**: False

**timedMetadataPid**

Packet Identifier (PID) of the timed metadata stream in the transport stream.

- **Type**: integer
- **Required**: False
- **Minimum**: 32
- **Maximum**: 8182

**scte35Source**

- **Type**: string
- **Required**: False

**pmtPid**

Packet Identifier (PID) for the Program Map Table (PMT) in the transport stream.

- **Type**: integer
- **Required**: False
- **Minimum**: 32
- **Maximum**: 8182

**bufferModel**

- **Type**: string
- **Required**: False

**ebpPlacement**

- **Type**: string
- **Required**: False

**dvbSdtSettings**

- **Type**: `DvbSdtSettings` (p. 106)
nullPacketBitrate

Value in bits per second of extra null packets to insert into the transport stream. This can be used if a downstream encryption system requires periodic null packets.

Type: number
Required: False
Format: float
Minimum: 0.0

pcrPid

Packet Identifier (PID) of the Program Clock Reference (PCR) in the transport stream. When no value is given, the encoder will assign the same value as the Video PID.

Type: integer
Required: False
Minimum: 32
Maximum: 8182

minEbpInterval

When set, enforces that Encoder Boundary Points do not come within the specified time interval of each other by looking ahead at input video. If another EBP is going to come in within the specified time interval, the current EBP is not emitted, and the segment is "stretched" to the next marker. The lookahead value does not add latency to the system. The Live Event must be configured elsewhere to create sufficient latency to make the lookahead accurate.

Type: integer
Required: False
Minimum: 0
Maximum: 10000

transportStreamId

The value of the transport stream ID field in the Program Map Table.

Type: integer
Required: False
Minimum: 0
Maximum: 65535

videoPid

Packet Identifier (PID) of the elementary video stream in the transport stream.

Type: integer
Required: False
Minimum: 32
Maximum: 8182
pcrControl

Type: string
Required: False

esRateInPes

Type: string
Required: False

segmentationMarkers

Type: string
Required: False

dvbTdtSettings

Type: DvbTdtSettings (p. 111)
Required: False

patInterval

The number of milliseconds between instances of this table in the output transport stream.

Type: integer
Required: False
Minimum: 0
Maximum: 1000

dvbSubPids

Packet Identifier (PID) for input source DVB Subtitle data to this output. Multiple values are accepted, and can be entered in ranges and/or by comma separation.

Type: Array of type integer
Required: False

M3u8NielsenId3 (enum)

If INSERT, Nielsen inaudible tones for media tracking will be detected in the input audio and an equivalent ID3 tag will be inserted in the output.

INSERT
NONE

M3u8PcrControl (enum)

When set to PCR_EVERY_PES_PACKET a Program Clock Reference value is inserted for every Packetized Elementary Stream (PES) header. This parameter is effective only when the PCR PID is the same as the video or audio elementary stream.

PCR_EVERY_PES_PACKET
CONFIGURED_PCR_PERIOD
M3u8Scte35Source (enum)

Enables SCTE-35 passthrough (scte35Source) to pass any SCTE-35 signals from input to output.

- PASSTHROUGH
- NONE

M3u8Settings

pmtPid

Packet Identifier (PID) for the Program Map Table (PMT) in the transport stream.

- Type: integer
- Required: False
- Minimum: 32
- Maximum: 8182

nielsenId3

- Type: string
- Required: False

pcrPid

Packet Identifier (PID) of the Program Clock Reference (PCR) in the transport stream. When no value is given, the encoder will assign the same value as the Video PID.

- Type: integer
- Required: False
- Minimum: 32
- Maximum: 8182

audioPids

Packet Identifier (PID) of the elementary audio stream(s) in the transport stream. Multiple values are accepted, and can be entered in ranges and/or by comma separation.

- Type: Array of type integer
- Required: False

audioFramesPerPes

The number of audio frames to insert for each PES packet.

- Type: integer
- Required: False
- Minimum: 0
- Maximum: 2147483647

scte35Pid

Packet Identifier (PID) of the SCTE-35 stream in the transport stream.
Properties

**Type**: integer  
**Required**: False  
**Minimum**: 32  
**Maximum**: 8182

**transportStreamId**
The value of the transport stream ID field in the Program Map Table.  
**Type**: integer  
**Required**: False  
**Minimum**: 0  
**Maximum**: 65535

**pcrControl**
**Type**: string  
**Required**: False

**videoPid**
Packet Identifier (PID) of the elementary video stream in the transport stream.  
**Type**: integer  
**Required**: False  
**Minimum**: 32  
**Maximum**: 8182

**privateMetadataPid**
Packet Identifier (PID) of the private metadata stream in the transport stream.  
**Type**: integer  
**Required**: False  
**Minimum**: 32  
**Maximum**: 8182

**pmtInterval**
The number of milliseconds between instances of this table in the output transport stream.  
**Type**: integer  
**Required**: False  
**Minimum**: 0  
**Maximum**: 1000

**patInterval**
The number of milliseconds between instances of this table in the output transport stream.  
**Type**: integer  
**Required**: False  
**Minimum**: 0
Maximum: 1000

**programNumber**

The value of the program number field in the Program Map Table.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
- **Maximum:** 65535

**timedMetadataPid**

Packet Identifier (PID) of the timed metadata stream in the transport stream.

- **Type:** integer
- **Required:** False
- **Minimum:** 32
- **Maximum:** 8182

**timedMetadata**

- **Type:** string
- **Required:** False

**scte35Source**

- **Type:** string
- **Required:** False

**MovClapAtom (enum)**

When enabled, include 'clap' atom if appropriate for the video output settings.

- INCLUDE
- EXCLUDE

**MovCslgAtom (enum)**

When enabled, file composition times will start at zero, composition times in the 'ctts' (composition time to sample) box for B-frames will be negative, and a 'cslg' (composition shift least greatest) box will be included per 14496-1 amendment 1. This improves compatibility with Apple players and tools.

- INCLUDE
- EXCLUDE

**MovMpeg2FourCCControl (enum)**

When set to XDCAM, writes MPEG2 video streams into the QuickTime file using XDCAM fourcc codes. This increases compatibility with Apple editors and players, but may decrease compatibility with other players. Only applicable when the video codec is MPEG2.

- XDCAM
MPEG

**MovPaddingControl (enum)**

If set to OMNEON, inserts Omneon-compatible padding

- OMNEON
- NONE

**MovReference (enum)**

A value of 'external' creates separate media files and the wrapper file (.mov) contains references to these media files. A value of 'self_contained' creates only a wrapper (.mov) file and this file contains all of the media.

- SELF_CONTAINED
- EXTERNAL

**MovSettings**

*reference*

- **Type**: string
- **Required**: False

*paddingControl*

- **Type**: string
- **Required**: False

*mpeg2FourCCControl*

- **Type**: string
- **Required**: False

*cslgAtom*

- **Type**: string
- **Required**: False

*clapAtom*

- **Type**: string
- **Required**: False

**Mp2Settings**

*channels*

Set Channels to specify the number of channels in this output audio track. Choosing Mono in the console will give you 1 output channel; choosing Stereo will give you 2. In the API, valid values are 1 and 2.
Properties

**bitrate**

Average bitrate in bits/second.

- **Type**: integer
- **Required**: False
- **Minimum**: 32000
- **Maximum**: 384000

**sampleRate**

Sample rate in hz.

- **Type**: integer
- **Required**: False
- **Minimum**: 32000
- **Maximum**: 48000

**Mp4CslgAtom (enum)**

When enabled, file composition times will start at zero, composition times in the 'ctts' (composition time to sample) box for B-frames will be negative, and a 'cslg' (composition shift least greatest) box will be included per 14496-1 amendment 1. This improves compatibility with Apple players and tools.

- **INCLUDE**
- **EXCLUDE**

**Mp4FreeSpaceBox (enum)**

Inserts a free-space box immediately after the moov box.

- **INCLUDE**
- **EXCLUDE**

**Mp4MoovPlacement (enum)**

If set to PROGRESSIVE_DOWNLOAD, the MOOV atom is relocated to the beginning of the archive as required for progressive downloading. Otherwise it is placed normally at the end.

- **PROGRESSIVE_DOWNLOAD**
- **NORMAL**

**Mp4Settings**

**mp4MajorBrand**

Overrides the "Major Brand" field in the output file. Usually not necessary to specify.
Properties

**Type**: string  
**Required**: False

**moovPlacement**

**Type**: string  
**Required**: False

**cslgAtom**

**Type**: string  
**Required**: False

**freeSpaceBox**

**Type**: string  
**Required**: False

**Mpeg2AdaptiveQuantization (enum)**  
Adaptive quantization. Allows intra-frame quantizers to vary to improve visual quality.

- OFF
- LOW
- MEDIUM
- HIGH

**Mpeg2CodecLevel (enum)**  
Use Level (Mpeg2CodecLevel) to set the MPEG-2 level for the video output.

- AUTO
- LOW
- MAIN
- HIGH1440
- HIGH

**Mpeg2CodecProfile (enum)**  
Use Profile (Mpeg2CodecProfile) to set the MPEG-2 profile for the video output.

- MAIN
- PROFILE_422

**Mpeg2FramerateControl (enum)**

If you are using the console, use the Framerate setting to specify the framerate for this output. If you want to keep the same framerate as the input video, choose Follow source. If you want to do framerate conversion, choose a framerate from the dropdown list or choose Custom. The frame rates shown in the dropdown list are decimal approximations of fractions. If you choose Custom, specify your framerate as a fraction. If you are creating your transcoding job specification as a JSON file without the console, use FramerateControl to specify which value the service uses for the framerate for this output. Choose INITIALIZE_FROM_SOURCE if you want the service to use the framerate from the input. Choose
SPECIFIED if you want the service to use the framerate you specify in the settings FramerateNumerator and FramerateDenominator.

INITIALIZE_FROM_SOURCE
SPECIFIED

**Mpeg2FramerateConversionAlgorithm (enum)**

When set to INTERPOLATE, produces smoother motion during framerate conversion.

DUPLICATE_DROP
INTERPOLATE

**Mpeg2GopSizeUnits (enum)**

Indicates if the GOP Size in MPEG2 is specified in frames or seconds. If seconds the system will convert the GOP Size into a frame count at run time.

FRAMES
SECONDS

**Mpeg2InterlaceMode (enum)**

Use Interlace mode (InterlaceMode) to choose the scan line type for the output. * Top Field First (TOP_FIELD) and Bottom Field First (BOTTOM_FIELD) produce interlaced output with the entire output having the same field polarity (top or bottom first). * Follow, Default Top (FOLLOW_TOP_FIELD) and Follow, Default Bottom (FOLLOW_BOTTOM_FIELD) use the same field polarity as the source. Therefore, behavior depends on the input scan type. - If the source is interlaced, the output will be interlaced with the same polarity as the source (it will follow the source). The output could therefore be a mix of "top field first" and "bottom field first". - If the source is progressive, the output will be interlaced with "top field first" or "bottom field first" polarity, depending on which of the Follow options you chose.

PROGRESSIVE
TOP_FIELD
BOTTOM_FIELD
FOLLOW_TOP_FIELD
FOLLOW_BOTTOM_FIELD

**Mpeg2IntraDcPrecision (enum)**

Use Intra DC precision (Mpeg2IntraDcPrecision) to set quantization precision for intra-block DC coefficients. If you choose the value auto, the service will automatically select the precision based on the per-frame compression ratio.

AUTO
INTRA_DC_PRECISION_8
INTRA_DC_PRECISION_9
INTRA_DC_PRECISION_10
INTRA_DC_PRECISION_11

**Mpeg2ParControl (enum)**

Using the API, enable ParFollowSource if you want the service to use the pixel aspect ratio from the input. Using the console, do this by choosing Follow source for Pixel aspect ratio.
INITIALIZE_FROM_SOURCE
SPECIFIED

**Mpeg2QualityTuningLevel (enum)**

Use Quality tuning level (Mpeg2QualityTuningLevel) to specify whether to use single-pass or multipass video encoding.

- SINGLE_PASS
- MULTI_PASS

**Mpeg2RateControlMode (enum)**

Use Rate control mode (Mpeg2RateControlMode) to specify whether the bitrate is variable (vbr) or constant (cbr).

- VBR
- CBR

**Mpeg2SceneChangeDetect (enum)**

Scene change detection (inserts I-frames on scene changes).

- DISABLED
- ENABLED

**Mpeg2Settings**

**minIInterval**

Enforces separation between repeated (cadence) I-frames and I-frames inserted by Scene Change Detection. If a scene change I-frame is within I-interval frames of a cadence I-frame, the GOP is shrunk and/or stretched to the scene change I-frame. GOP stretch requires enabling lookahead as well as setting I-interval. The normal cadence resumes for the next GOP. This setting is only used when Scene Change Detect is enabled. Note: Maximum GOP stretch = GOP size + Min-I-interval - 1

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 30

**parNumerator**

Pixel Aspect Ratio numerator.

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 2147483647

**gopSizeUnits**

- **Type**: string
**Required:** False

**hrdBufferSize**

Size of buffer (HRD buffer model) in bits. For example, enter five megabits as 5000000.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
- **Maximum:** 47185920

**qualityTuningLevel**

- **Type:** string
- **Required:** False

**maxBitrate**

Maximum bitrate in bits/second. For example, enter five megabits per second as 5000000.

- **Type:** integer
- **Required:** False
- **Minimum:** 1000
- **Maximum:** 300000000

**bitrate**

Average bitrate in bits/second. Required for VBR and CBR. For MS Smooth outputs, bitrates must be unique when rounded down to the nearest multiple of 1000.

- **Type:** integer
- **Required:** False
- **Minimum:** 1000
- **Maximum:** 288000000

**spatialAdaptiveQuantization**

- **Type:** string
- **Required:** False

**slowPal**

- **Type:** string
- **Required:** False

**intraDcPrecision**

- **Type:** string
- **Required:** False

**codecProfile**

- **Type:** string
Properties

**softness**

Softness. Selects quantizer matrix, larger values reduce high-frequency content in the encoded image.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
- **Maximum:** 128

**framerateControl**

- **Type:** string
- **Required:** False

**telecine**

- **Type:** string
- **Required:** False

**codecLevel**

- **Type:** string
- **Required:** False

**framerateConversionAlgorithm**

- **Type:** string
- **Required:** False

**temporalAdaptiveQuantization**

- **Type:** string
- **Required:** False

**hrdBufferInitialFillPercentage**

Percentage of the buffer that should initially be filled (HRD buffer model).

- **Type:** integer
- **Required:** False
- **Minimum:** 0
- **Maximum:** 100

**framerateNumerator**

Framerate numerator - framerate is a fraction, e.g. 24000 / 1001 = 23.976 fps.

- **Type:** integer
- **Required:** False
- **Minimum:** 24
Maximum: 60000

**gopClosedCadence**

Frequency of closed GOPs. In streaming applications, it is recommended that this be set to 1 so a decoder joining mid-stream will receive an IDR frame as quickly as possible. Setting this value to 0 will break output segmenting.

Type: integer
Required: False
Minimum: 0
Maximum: 2147483647

**numberBFramesBetweenReferenceFrames**

Number of B-frames between reference frames.

Type: integer
Required: False
Minimum: 0
Maximum: 7

**framerateDenominator**

Framerate denominator.

Type: integer
Required: False
Minimum: 1
Maximum: 1001

**adaptiveQuantization**

Type: string
Required: False

**interlaceMode**

Type: string
Required: False

**gopSize**

GOP Length (keyframe interval) in frames or seconds. Must be greater than zero.

Type: number
Required: False
Format: float
Minimum: 0.0

**syntax**

Type: string
Required: False

parDenominator

Pixel Aspect Ratio denominator.

Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

sceneChangeDetect

Type: string
Required: False

parControl

Type: string
Required: False

rateControlMode

Type: string
Required: False

Mpeg2SlowPal (enum)

Enables Slow PAL rate conversion. 23.976fps and 24fps input is relabeled as 25fps, and audio is sped up correspondingly.

DISABLED
ENABLED

Mpeg2SpatialAdaptiveQuantization (enum)

Adjust quantization within each frame based on spatial variation of content complexity.

DISABLED
ENABLED

Mpeg2Syntax (enum)

Produces a Type D-10 compatible bitstream (SMPTE 356M-2001).

DEFAULT
D_10

Mpeg2Telecine (enum)

Only use Telecine (Mpeg2Telecine) when you set Framerate (Framerate) to 29.970. Set Telecine (Mpeg2Telecine) to Hard (hard) to produce a 29.97i output from a 23.976 input. Set it to Soft (soft) to produce 23.976 output and leave conversion to the player.
Properties

NONE
SOFT
HARD

**Mpeg2TemporalAdaptiveQuantization** (enum)

Adjust quantization within each frame based on temporal variation of content complexity.

- DISABLED
- ENABLED

**MsSmoothAudioDeduplication** (enum)

COMBINE_DUPLICATE_STREAMS combines identical audio encoding settings across a Microsoft Smooth output group into a single audio stream.

- COMBINE_DUPLICATE_STREAMS
- NONE

**MsSmoothEncryptionSettings**

**spekeKeyProvider**

- **Type**: SpekeKeyProvider (p. 189)
- **Required**: True

**MsSmoothGroupSettings**

**fragmentLength**

Use Fragment length (FragmentLength) to specify the mp4 fragment sizes in seconds. Fragment length must be compatible with GOP size and framerate.

- **Type**: integer
- **Required**: True
- **Minimum**: 1
- **Maximum**: 2147483647

**encryption**

- **Type**: MsSmoothEncryptionSettings (p. 178)
- **Required**: False

**audioDeduplication**

- **Type**: string
- **Required**: False

**manifestEncoding**

- **Type**: string
- **Required**: False
**destination**

Use Destination (Destination) to specify the S3 output location and the output filename base. Destination accepts format identifiers. If you do not specify the base filename in the URI, the service will use the filename of the input file. If your job has multiple inputs, the service uses the filename of the first input file.

- **Type:** string
- **Required:** False
- **Pattern:** ^s3:\/\(/

**MsSmoothManifestEncoding (enum)**

Use Manifest encoding (MsSmoothManifestEncoding) to specify the encoding format for the server and client manifest. Valid options are utf8 and utf16.

- UTF8
- UTF16

**NielsenConfiguration**

**distributorId**

Use Distributor ID (DistributorID) to specify the distributor ID that is assigned to your organization by Nielsen.

- **Type:** string
- **Required:** False

**breakoutCode**

Use Nielsen Configuration (NielsenConfiguration) to set the Nielsen measurement system breakout code. Supported values are 0, 3, 7, and 9.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
- **Maximum:** 9

**NoiseReducer**

**filter**

- **Type:** string
- **Required:** True

**filterSettings**

- **Type:** NoiseReducerFilterSettings (p. 180)
- **Required:** False

**spatialFilterSettings**

- **Type:** NoiseReducerSpatialFilterSettings (p. 180)
Required: False

**NoiseReducerFilter (enum)**

Use Noise reducer filter (NoiseReducerFilter) to select one of the following spatial image filtering functions. To use this setting, you must also enable Noise reducer (NoiseReducer). * Bilateral is an edge preserving noise reduction filter. * Mean (softest), Gaussian, Lanczos, and Sharpen (sharpest) are convolution filters. * Conserve is a min/max noise reduction filter. * Spatial is a frequency-domain filter based on JND principles.

- BILATERAL
- MEAN
- GAUSSIAN
- LANCZOS
- SHARPEN
- CONSERVE
- SPATIAL

**NoiseReducerFilterSettings**

**strength**

Relative strength of noise reducing filter. Higher values produce stronger filtering.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 3

**NoiseReducerSpatialFilterSettings**

**strength**

Relative strength of noise reducing filter. Higher values produce stronger filtering.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 16

**postFilterSharpenStrength**

Specify strength of post noise reduction sharpening filter, with 0 disabling the filter and 3 enabling it at maximum strength.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 3

**speed**

The speed of the filter, from -2 (lower speed) to 3 (higher speed), with 0 being the nominal value.

- **Type**: integer
Properties

**Required**: False
**Minimum**: -2
**Maximum**: 3

**Order (enum)**

When you request lists of resources, you can optionally specify whether they are sorted in ASCENDING or DESCENDING order. Default varies by resource.

- ASCENDING
- DESCENDING

**Output**

**extension**

Use Extension (Extension) to specify the file extension for outputs in File output groups. If you do not specify a value, the service will use default extensions by container type as follows:
- MPEG-2 transport stream, m2ts
- Quicktime, mov
- MXF container, mxf
- MPEG-4 container, mp4
- No Container, the service will use codec extensions (e.g. AAC, H265, H266, AC3)

- **Type**: string
- **Required**: False

**videoDescription**

(VideoDescription) contains a group of video encoding settings. The specific video settings depend on the video codec you choose when you specify a value for Video codec (codec). Include one instance of (VideoDescription) per output.

- **Type**: VideoDescription (p. 194)
- **Required**: False

**audioDescriptions**

(AudioDescriptions) contains groups of audio encoding settings organized by audio codec. Include one instance of (AudioDescriptions) per output. (AudioDescriptions) can contain multiple groups of encoding settings.

- **Type**: Array of type AudioDescription (p. 85)
- **Required**: False

**containerSettings**

- **Type**: ContainerSettings (p. 101)
- **Required**: False

**preset**

Use Preset (Preset) to specify a preset for your transcoding settings. Provide the system or custom preset name. You can specify either Preset (Preset) or Container settings (ContainerSettings), but not both.

- **Type**: string
- **Required**: False
outputSettings
Type: OutputSettings (p. 184)
Required: False

captionDescriptions

(CaptionDescriptions) contains groups of captions settings. For each output that has captions, include one instance of (CaptionDescriptions). (CaptionDescriptions) can contain multiple groups of captions settings.

Type: Array of type CaptionDescription (p. 92)
Required: False

nameModifier

Use Name modifier (NameModifier) to have the service add a string to the end of each output filename. You specify the base filename as part of your destination URI. When you create multiple outputs in the same output group, Name modifier (NameModifier) is required. Name modifier also accepts format identifiers. For DASH ISO outputs, if you use the format identifiers $Number$ or $Time$ in one output, you must use them in the same way in all outputs of the output group.

Type: string
Required: False

OutputChannelMapping

inputChannels

List of input channels

Type: Array of type integer
Required: True

OutputGroup

outputs

This object holds groups of encoding settings, one group of settings per output.

Type: Array of type Output (p. 181)
Required: True

outputGroupSettings

Type: OutputGroupSettings (p. 183)
Required: True

name

Name of the output group

Type: string
Required: False

customName

Use Custom Group Name (CustomName) to specify a name for the output group. This value is displayed on the console and can make your job settings JSON more human-readable. It does not affect your outputs. Use up to twelve characters that are either letters, numbers, spaces, or underscores.

Type: string
Required: False

OutputGroupSettings

dashIsoGroupSettings

Type: DashIsoGroupSettings (p. 103)
Required: False

fileGroupSettings

Type: FileGroupSettings (p. 117)
Required: False

msSmoothGroupSettings

Type: MsSmoothGroupSettings (p. 178)
Required: False

cmafGroupSettings

Type: CmafGroupSettings (p. 96)
Required: False

type

Type: string
Required: True

hlsGroupSettings

Type: HlsGroupSettings (p. 141)
Required: False

OutputGroupType (enum)

Type of output group (File group, Apple HLS, DASH ISO, Microsoft Smooth Streaming, CMAF)

HLS_GROUP_SETTINGS
DASH_ISO_GROUP_SETTINGS
FILE_GROUP_SETTINGS
MS_SMOOTH_GROUP_SETTINGS
CMAF_GROUP_SETTINGS
**OutputSdt (enum)**

Selects method of inserting SDT information into output stream. "Follow input SDT" copies SDT information from input stream to output stream. "Follow input SDT if present" copies SDT information from input stream to output stream if SDT information is present in the input, otherwise it will fall back on the user-defined values. Enter "SDT Manually" means user will enter the SDT information. "No SDT" means output stream will not contain SDT information.

- SDT_FOLLOW
- SDT_FOLLOW_IF_PRESENT
- SDT_MANUAL
- SDT_NONE

**OutputSettings**

**hlsSettings**

*Type: HlsSettings (p. 145)*

*Required: False*

**ProresCodecProfile (enum)**

Use Profile (ProResCodecProfile) to specify the type of Apple ProRes codec to use for this output.

- APPLE_PRORES_422
- APPLE_PRORES_422_HQ
- APPLE_PRORES_422_LT
- APPLE_PRORES_422_PROXY

**ProresFramerateControl (enum)**

If you are using the console, use the Framerate setting to specify the framerate for this output. If you want to keep the same framerate as the input video, choose Follow source. If you want to do framerate conversion, choose a framerate from the dropdown list or choose Custom. The framerates shown in the dropdown list are decimal approximations of fractions. If you choose Custom, specify your framerate as a fraction. If you are creating your transcoding job specification as a JSON file without the console, use FramerateControl to specify which value the service uses for the framerate for this output. Choose INITIALIZE_FROM_SOURCE if you want the service to use the framerate from the input. Choose SPECIFIED if you want the service to use the framerate you specify in the settings FramerateNumerator and FramerateDenominator.

- INITIALIZE_FROM_SOURCE
- SPECIFIED

**ProresFramerateConversionAlgorithm (enum)**

When set to INTERPOLATE, produces smoother motion during framerate conversion.

- DUPLICATE_DROP
- INTERPOLATE

**ProresInterlaceMode (enum)**

Use Interlace mode (InterlaceMode) to choose the scan line type for the output. * Top Field First (TOP_FIELD) and Bottom Field First (BOTTOM_FIELD) produce interlaced output with the entire output
having the same field polarity (top or bottom first). * Follow, Default Top (FOLLOW_TOP_FIELD) and Follow, Default Bottom (FOLLOW_BOTTOM_FIELD) use the same field polarity as the source. Therefore, behavior depends on the input scan type. - If the source is interlaced, the output will be interlaced with the same polarity as the source (it will follow the source). The output could therefore be a mix of “top field first” and “bottom field first”. - If the source is progressive, the output will be interlaced with “top field first” or “bottom field first” polarity, depending on which of the Follow options you chose.

**PROGRESSIVE**
- TOP_FIELD
- BOTTOM_FIELD
- FOLLOW_TOP_FIELD
- FOLLOW_BOTTOM_FIELD

**ProresParControl (enum)**

Use (ProresParControl) to specify how the service determines the pixel aspect ratio. Set to Follow source (INITIALIZE_FROM_SOURCE) to use the pixel aspect ratio from the input. To specify a different pixel aspect ratio: Using the console, choose it from the dropdown menu. Using the API, set ProresParControl to (SPECIFIED) and provide for (ParNumerator) and (ParDenominator).

- INITIALIZE_FROM_SOURCE
- SPECIFIED

**ProresSettings**

**framerateDenominator**

Framerate denominator.

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 2147483647

**slowPal**

- **Type**: string
- **Required**: False

**framerateControl**

- **Type**: string
- **Required**: False

**telecine**

- **Type**: string
- **Required**: False

**framerateConversionAlgorithm**

- **Type**: string
**Properties**

**Required**: False

**interlaceMode**

*Type*: string  
*Required*: False

**parNumerator**

Pixel Aspect Ratio numerator.

*Type*: integer  
*Required*: False  
*Minimum*: 1  
*Maximum*: 2147483647

**codecProfile**

*Type*: string  
*Required*: False

**parDenominator**

Pixel Aspect Ratio denominator.

*Type*: integer  
*Required*: False  
*Minimum*: 1  
*Maximum*: 2147483647

**parControl**

*Type*: string  
*Required*: False

**framerateNumerator**

When you use the API for transcode jobs that use framerate conversion, specify the framerate as a fraction. For example, \( \frac{24000}{1001} = 23.976 \) fps. Use FramerateNumerator to specify the numerator of this fraction. In this example, use 24000 for the value of FramerateNumerator.

*Type*: integer  
*Required*: False  
*Minimum*: 1  
*Maximum*: 2147483647

**ProresSlowPal (enum)**

Enables Slow PAL rate conversion. \( 23.976 \text{fps} \) and \( 24 \text{fps} \) input is relabeled as \( 25 \text{fps} \), and audio is sped up correspondingly.

*DISABLED*  
*ENABLED*
ProresTelecine (enum)

Only use Telecine (ProresTelecine) when you set Framerate (Framerate) to 29.970. Set Telecine (ProresTelecine) to Hard (hard) to produce a 29.97i output from a 23.976 input. Set it to Soft (soft) to produce 23.976 output and leave conversion to the player.

- NONE
- HARD

Rectangle

width

Width of rectangle in pixels.

- Type: integer
- Required: True
- Minimum: -2147483648
- Maximum: 2147483647

x

The distance, in pixels, between the rectangle and the left edge of the video frame.

- Type: integer
- Required: True
- Minimum: -2147483648
- Maximum: 2147483647

y

The distance, in pixels, between the rectangle and the top edge of the video frame.

- Type: integer
- Required: True
- Minimum: -2147483648
- Maximum: 2147483647

height

Height of rectangle in pixels.

- Type: integer
- Required: True
- Minimum: -2147483648
- Maximum: 2147483647

RemixSettings

channelsOut

Specify the number of channels in this output after remixing. Valid values: 1, 2, 4, 6, 8

- Type: integer
Required: True
Minimum: 1
Maximum: 8

channelMapping

Type: ChannelMapping (p. 95)
Required: True

channelsIn

Specify the number of audio channels from your input that you want to use in your output. With
remixing, you might combine or split the data in these channels, so the number of channels in your final
output might be different.

Type: integer
Required: True
Minimum: 1
Maximum: 16

RespondToAfd (enum)

Use Respond to AFD (RespondToAfd) to specify how the service changes the video itself in response
to AFD values in the input. * Choose Respond to clip the input video frame according to the AFD value,
input display aspect ratio, and output display aspect ratio. * Choose Passthrough to include the input
AFD values. Do not choose this when AfdSignaling is set to (NONE). A preferred implementation of this
workflow is to set RespondToAfd to (NONE) and set AfdSignaling to (AUTO). * Choose None to remove all
input AFD values from this output.

NONE
RESPOND
PASSTHROUGH

ScalingBehavior (enum)

Applies only if your input aspect ratio is different from your output aspect ratio. Enable Stretch to output
(StretchToOutput) to have the service stretch your video image to fit. Leave this setting disabled to
allow the service to letterbox your video instead. This setting overrides any positioning value you specify
elsewhere in the job.

DEFAULT
STRETCH_TO_OUTPUT

SccDestinationFramerate (enum)

Set Framerate (SccDestinationFramerate) to make sure that the captions and the video are synchronized
in the output. Specify a framerate that matches the framerate of the associated video. If the video
framerate is 29.97, choose 29.97 dropframe (FRAMERATE_29_97_DROPFRAME) only if the video
has video_insertion=true and drop_frame_timecode=true; otherwise, choose 29.97 non-dropframe
(FRAMERATE_29_97_NON_DROPFRAME).

FRAMERATE_23_97
FRAMERATE_24
FRAMERATE_29_97_DROPFRAME
FRAMERATE_29_97_NON_DROPFRAME
**SccDestinationSettings**

**framerate**
- **Type:** string
- **Required:** False

**SpekeKeyProvider**

**resourceId**
The SPEKE-compliant server uses Resource ID (ResourceId) to identify content.
- **Type:** string
- **Required:** True
- **Pattern:** `^[\w-]+$`

**systemIds**
Relates to SPEKE implementation. DRM system identifiers. DASH output groups support a max of two system ids. Other group types support one system id.
- **Type:** Array of type string
- **Required:** True

**url**
Use URL (Url) to specify the SPEKE-compliant server that will provide keys for content.
- **Type:** string
- **Required:** True
- **Format:** uri
- **Pattern:** `^https:/\//`

**StaticKeyProvider**

**keyFormatVersions**
Relates to DRM implementation. Either a single positive integer version value or a slash delimited list of version values (1/2/3).
- **Type:** string
- **Required:** False
- **Pattern:** `^\d+(\d*+)$`

**keyFormat**
Relates to DRM implementation. Sets the value of the KEYFORMAT attribute. Must be 'identity' or a reverse DNS string. May be omitted to indicate an implicit value of 'identity'.
- **Type:** string
- **Required:** False
Properties

**Pattern**

Pattern: `^(identity|[A-Za-z]{2,6}(\.[A-Za-z0-9-]{1,63})+)$`

**staticKeyValue**

Relates to DRM implementation. Use a 32-character hexadecimal string to specify Key Value (StaticKeyValue).

- **Type**: string
- **Required**: True
- **Pattern**: `^[A-Za-z0-9]{32}$`

**url**

Relates to DRM implementation. The location of the license server used for protecting content.

- **Type**: string
- **Required**: True
- **Format**: uri

**TeletextDestinationSettings**

**pageNumber**

Set pageNumber to the Teletext page number for the destination captions for this output. This value must be a three-digit hexadecimal string; strings ending in -FF are invalid. If you are passing through the entire set of Teletext data, do not use this field.

- **Type**: string
- **Required**: False
- **Pattern**: `^[1-8][0-9a-fA-F][0-9a-eA-E]$`

**TeletextSourceSettings**

**pageNumber**

Use Page Number (PageNumber) to specify the three-digit hexadecimal page number that will be used for Teletext captions. Do not use this setting if you are passing through teletext from the input source to output.

- **Type**: string
- **Required**: False
- **Pattern**: `^[1-8][0-9a-fA-F][0-9a-eA-E]$`

**TimecodeBurnin**

**prefix**

Use Prefix (Prefix) to place ASCII characters before any burned-in timecode. For example, a prefix of "EZ-" will result in the timecode "EZ-00:00:00:00". Provide either the characters themselves or the ASCII code equivalents. The supported range of characters is 0x20 through 0x7e. This includes letters, numbers, and all special characters represented on a standard English keyboard.

- **Type**: string
- **Required**: False
Properties

Pattern

Use Font Size (FontSize) to set the font size of any burned-in timecode. Valid values are 10, 16, 32, 48.

fontSize

- **Type**: integer
- **Required**: False
- **Minimum**: 10
- **Maximum**: 48

position

- **Type**: string
- **Required**: False

TimecodeBurninPosition (enum)

Use Position (Position) under Timecode burn-in (TimecodeBurnIn) to specify the location the burned-in timecode on output video.

- TOP_CENTER
- TOP_LEFT
- TOP_RIGHT
- MIDDLE_LEFT
- MIDDLE_CENTER
- MIDDLE_RIGHT
- BOTTOM_LEFT
- BOTTOM_CENTER
- BOTTOM_RIGHT

TimecodeConfig

timestampOffset

Only applies to outputs that support program-date-time stamp. Use Timestamp offset (TimestampOffset) to overwrite the timecode date without affecting the time and frame number. Provide the new date as a string in the format "yyyy-mm-dd". To use Time stamp offset, you must also enable Insert program-date-time (InsertProgramDateTime) in the output settings. For example, if the date part of your timecodes is 2002-1-25 and you want to change it to one year later, set Timestamp offset (TimestampOffset) to 2003-1-25.

- **Type**: string
- **Required**: False
- **Pattern**: ^([0-9]{4})-(0[1-9]|1[0-2])-(0[1-9]|[12][0-9]|3[01])$

anchor

If you use an editing platform that relies on an anchor timecode, use Anchor Timecode (Anchor) to specify a timecode that will match the input video frame to the output video frame. Use 24-hour format with frame number, (HH:MM:SS:FF) or (HH:MM:SS:FF). This setting ignores framerate conversion. System behavior for Anchor Timecode varies depending on your setting for Source (TimecodeSource).

- If Source (TimecodeSource) is set to Specified Start (SPECIFIEDSTART), the first input frame is the specified value in Start Timecode (Start). Anchor Timecode (Anchor) and Start Timecode (Start) are used
calculate output timecode. * If Source (TimecodeSource) is set to Start at 0 (ZEROBASED) the first frame is 00:00:00:00. * If Source (TimecodeSource) is set to Embedded (EMBEDDED), the first frame is the timecode value on the first input frame of the input.

Type: string  
Required: False  
Format: timecode  
Pattern: ^([01][0-9]|2[0-4]):[0-5][0-9]:[0-5][0-9]:[0-9]{2}$

start

Only use when you set Source (TimecodeSource) to Specified start (SPECIFIEDSTART). Use Start timecode (Start) to specify the timecode for the initial frame. Use 24-hour format with frame number, (HH:MM:SS:FF) or (HH:MM:SS;FF).

Type: string  
Required: False  
Format: timecode  
Pattern: ^([01][0-9]|2[0-4]):[0-5][0-9]:[0-5][0-9]:[0-9]{2}$

source

Type: string  
Required: False

TimecodeSource (enum)

Use Source (TimecodeSource) to set how timecodes are handled within this job. To make sure that your video, audio, captions, and markers are synchronized and that time-based features, such as image inserter, work correctly, choose the Timecode source option that matches your assets. All timecodes are in a 24-hour format with frame number (HH:MM:SS:FF). * Embedded (EMBEDDED) - Use the timecode that is in the input video. If no embedded timecode is in the source, the service will use Start at 0 (ZEROBASED) instead. * Start at 0 (ZEROBASED) - Set the timecode of the initial frame to 00:00:00:00. * Specified Start (SPECIFIEDSTART) - Set the timecode of the initial frame to a value other than zero. You use Start timecode (Start) to provide this value.

EMBEDDED  
ZEROBASED  
SPECIFIEDSTART

TimedMetadata (enum)

Applies only to HLS outputs. Use this setting to specify whether the service inserts the ID3 timed metadata from the input in this output.

PASSTHROUGH  
NONE

TimedMetadataInsertion

id3Insertions

Id3Insertions contains the array of Id3Insertion instances.

Type: Array of type Id3Insertion (p. 146)
**Required**: True

**TtmlDestinationSettings**

**stylePassthrough**

- **Type**: string
- **Required**: False

**TtmlStylePassthrough (enum)**

Pass through style and position information from a TTML-like input source (TTML, SMPTE-TT, CFF-TT) to the CFF-TT output or TTML output.

- ENABLED
- DISABLED

**Type (enum)**

- SYSTEM
- CUSTOM

**VideoCodec (enum)**

Type of video codec

- FRAME_CAPTURE
- H_264
- H_265
- MPEG2
- PRORES

**VideoCodecSettings**

**h265Settings**

- **Type**: H265Settings (p. 130)
- **Required**: False

**codec**

- **Type**: string
- **Required**: True

**proresSettings**

- **Type**: ProresSettings (p. 185)
- **Required**: False

**mpeg2Settings**

- **Type**: Mpeg2Settings (p. 173)
Properties

Required: False

frameCaptureSettings

Type: FrameCaptureSettings (p. 118)
Required: False

h264Settings

Type: H264Settings (p. 121)
Required: False

VideoDescription

fixedAfd
Applies only if you set AFD Signaling(AfdSignaling) to Fixed (FIXED). Use Fixed (FixedAfd) to specify a four-bit AFD value which the service will write on all frames of this video output.

Type: integer
Required: False
Minimum: 0
Maximum: 15

scalingBehavior

Type: string
Required: False

respondToAfd

Type: string
Required: False

codecSettings

Type: VideoCodecSettings (p. 193)
Required: True

afdSignaling

Type: string
Required: False

colorMetadata

Type: string
Required: False

timecodeInsertion

Type: string
Required: False

width
Use Width (Width) to define the video resolution width, in pixels, for this output. If you don't provide a value here, the service will use the input width.

Type: integer
Required: False
Minimum: 32
Maximum: 4096

videoPreprocessors
Find additional transcoding features under Preprocessors (VideoPreprocessors). Enable the features at each output individually. These features are disabled by default.

Type: VideoPreprocessor (p. 196)
Required: False

antiAlias

Type: string
Required: False

position
Use Position (Position) to point to a rectangle object to define your position. This setting overrides any other aspect ratio.

Type: Rectangle (p. 187)
Required: False

sharpness
Use Sharpness (Sharpness) setting to specify the strength of anti-aliasing. This setting changes the width of the anti-alias filter kernel used for scaling. Sharpness only applies if your output resolution is different from your input resolution, and if you set Anti-alias (AntiAlias) to ENABLED. 0 is the softest setting, 100 the sharpest, and 50 recommended for most content.

Type: integer
Required: False
Minimum: 0
Maximum: 100

crop
Applies only if your input aspect ratio is different from your output aspect ratio. Use Input cropping rectangle (Crop) to specify the video area the service will include in the output. This will crop the input source, causing video pixels to be removed on encode. Do not use this setting if you have enabled Stretch to output (stretchToOutput) in your output settings.

Type: Rectangle (p. 187)
Required: False
**dropFrameTimecode**

*Type:* string  
*Required:* False

**height**

Use the Height (Height) setting to define the video resolution height for this output. Specify in pixels. If you don't provide a value here, the service will use the input height.

*Type:* integer  
*Required:* False  
*Minimum:* 32  
*Maximum:* 2160

**VideoPreprocessor**

**noiseReducer**

Enable the Noise reducer (NoiseReducer) feature to remove noise from your video output if necessary. Enable or disable this feature for each output individually. This setting is disabled by default.

*Type:* NoiseReducer (p. 179)  
*Required:* False

**timecodeBurnin**

Timecode burn-in (TimecodeBurnIn)--Burns the output timecode and specified prefix into the output.

*Type:* TimecodeBurnin (p. 190)  
*Required:* False

**colorCorrector**

Enable the Color corrector (ColorCorrector) feature if necessary. Enable or disable this feature for each output individually. This setting is disabled by default.

*Type:* ColorCorrector (p. 99)  
*Required:* False

**deinterlacer**

Use Deinterlacer (Deinterlacer) to produce smoother motion and a clearer picture.

*Type:* Deinterlacer (p. 105)  
*Required:* False

**imageInserter**

Enable the Image inserter (ImageInserter) feature to include a graphic overlay on your video. Enable or disable this feature for each output individually. This setting is disabled by default.

*Type:* ImageInserter (p. 146)
**VideoSelector**

**colorSpace**

- **Type**: string
- **Required**: False

**hdr10Metadata**

- **Type**:Hdr10Metadata (p. 136)
- **Required**: False

**programNumber**

Selects a specific program from within a multi-program transport stream. Note that Quad 4K is not currently supported.

- **Type**: integer
- **Required**: False
- **Minimum**: -2147483648
- **Maximum**: 2147483647

**pid**

Use PID (Pid) to select specific video data from an input file. Specify this value as an integer; the system automatically converts it to the hexadecimal value. For example, 257 selects PID 0x101. A PID, or packet identifier, is an identifier for a set of data in an MPEG-2 transport stream container.

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 2147483647

**colorSpaceUsage**

- **Type**: string
- **Required**: False

**VideoTimecodeInsertion (enum)**

Applies only to H.264, H.265, MPEG2, and ProRes outputs. Only enable Timecode insertion when the input framerate is identical to the output framerate. To include timecodes in this output, set Timecode insertion (VideoTimecodeInsertion) to PIC_TIMING_SEI. To leave them out, set it to DISABLED. Default is DISABLED. When the service inserts timecodes in an output, by default, it uses any embedded timecodes from the input. If none are present, the service will set the timecode for the first output frame to zero. To change this default behavior, adjust the settings under Timecode configuration (TimecodeConfig). In the console, these settings are located under Job > Job settings > Timecode configuration. Note - Timecode source under input settings (InputTimecodeSource) does not affect the timecodes that are inserted in the output. Source under Job settings > Timecode configuration (TimecodeSource) does.

- **DISABLED**
PIC_TIMING_SEI

WavFormat (enum)

The service defaults to using RIFF for WAV outputs. If your output audio is likely to exceed 4 GB in file size, or if you otherwise need the extended support of the RF64 format, set your output WAV file format to RF64.

RIFF
RF64

WavSettings

channels

Set Channels to specify the number of channels in this output audio track. With WAV, valid values 1, 2, 4, and 8. In the console, these values are Mono, Stereo, 4-Channel, and 8-Channel, respectively.

Type: integer
Required: False
Minimum: 1
Maximum: 8

bitDepth

Specify Bit depth (BitDepth), in bits per sample, to choose the encoding quality for this audio track.

Type: integer
Required: False
Minimum: 16
Maximum: 24

format

Type: string
Required: False

sampleRate

Sample rate in Hz.

Type: integer
Required: False
Minimum: 8000
Maximum: 192000

JobTemplates name

URI

/2017-08-29/jobTemplates/ name
HTTP Methods

GET

Operation ID: GetJobTemplate

Retrieve the JSON for a specific job template.

Path Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>True</td>
<td></td>
</tr>
</tbody>
</table>

Responses

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>GetJobTemplateResponse (p. 211)</td>
<td>200 response</td>
</tr>
<tr>
<td>400</td>
<td>ExceptionBody (p. 232)</td>
<td>BadRequestException</td>
</tr>
<tr>
<td>500</td>
<td>ExceptionBody (p. 232)</td>
<td>InternalServiceException</td>
</tr>
<tr>
<td>403</td>
<td>ExceptionBody (p. 232)</td>
<td>AccessDeniedException</td>
</tr>
<tr>
<td>404</td>
<td>ExceptionBody (p. 232)</td>
<td>ResourceNotFoundException</td>
</tr>
<tr>
<td>429</td>
<td>ExceptionBody (p. 232)</td>
<td>LimitExceededException</td>
</tr>
<tr>
<td>409</td>
<td>ExceptionBody (p. 232)</td>
<td>ResourceInUseException</td>
</tr>
</tbody>
</table>

PUT

Operation ID: UpdateJobTemplate

Modify one of your existing job templates.

Path Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>True</td>
<td></td>
</tr>
</tbody>
</table>

Responses

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>UpdateJobTemplateResponse (p. 221)</td>
<td>200 response</td>
</tr>
<tr>
<td>400</td>
<td>ExceptionBody (p. 232)</td>
<td>BadRequestException</td>
</tr>
<tr>
<td>500</td>
<td>ExceptionBody (p. 232)</td>
<td>InternalServiceException</td>
</tr>
<tr>
<td>403</td>
<td>ExceptionBody (p. 232)</td>
<td>AccessDeniedException</td>
</tr>
<tr>
<td>404</td>
<td>ExceptionBody (p. 232)</td>
<td>ResourceNotFoundException</td>
</tr>
</tbody>
</table>
**DELETE**

Operation ID: DeleteJobTemplate

Permanently delete a job template you have created.

**Path Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>True</td>
<td></td>
</tr>
</tbody>
</table>

**Responses**

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>ExceptionBody (p. 232)</td>
<td>BadRequestException</td>
</tr>
<tr>
<td>202</td>
<td>DeleteJobTemplateResponse</td>
<td>202 response</td>
</tr>
<tr>
<td>500</td>
<td>ExceptionBody (p. 232)</td>
<td>InternalServiceException</td>
</tr>
<tr>
<td>403</td>
<td>ExceptionBody (p. 232)</td>
<td>AccessDeniedException</td>
</tr>
<tr>
<td>404</td>
<td>ExceptionBody (p. 232)</td>
<td>ResourceNotFoundException</td>
</tr>
<tr>
<td>429</td>
<td>ExceptionBody (p. 232)</td>
<td>LimitExceededException</td>
</tr>
<tr>
<td>409</td>
<td>ExceptionBody (p. 232)</td>
<td>ResourceInUseException</td>
</tr>
</tbody>
</table>

**Schemas**

**Request Bodies**

**Example GET**

```json
{
   "name (p. 271)": "string"
}
```

**Example PUT**

```json
{
   "settings (p. 345)": {
      "timecodeConfig (p. 306)": {
         "timestampOffset (p. 343)": "string",
         "anchor (p. 343)": "string",
         "start (p. 343)": "string",
      }
   }
}```
"source (p. 344)": enum
},
"adAvailOffset (p. 306)": integer,
"outputGroups (p. 306)": [
  
  "outputs (p. 334)": [
  
  "extension (p. 332)": "string",
  "videoDescription (p. 333)": {
    "fixedAfd (p. 346)": integer,
    "scalingBehavior (p. 346)": enum,
    "respondToAfd (p. 347)": enum,
    "codecSettings (p. 347)": {
      "h265Settings (p. 346)": {
        "slices (p. 283)": integer,
        "minIInterval (p. 283)": integer,
        "parNumerator (p. 283)": integer,
        "flickerAdaptiveQuantization (p. 283)": enum,
        "gopSizeUnits (p. 283)": enum,
        "hrdBufferSize (p. 284)": integer,
        "qualityTuningLevel (p. 284)": enum,
        "maxBitrate (p. 284)": integer,
        "bitrate (p. 284)": integer,
        "spatialAdaptiveQuantization (p. 284)": enum,
        "sampleAdaptiveOffsetFilterMode (p. 284)": enum,
        "temporalAdaptiveQuantization (p. 284)": enum,
        "framerateConversionAlgorithm (p. 285)": enum,
        "numberReferenceFrames (p. 285)": integer,
        "temporalAdaptiveQuantization (p. 286)": enum,
        "framerateConversionAlgorithm (p. 286)": enum,
        "numberBFramesBetweenReferenceFrames (p. 286)": integer,
        "framerateDenominator (p. 286)": integer,
        "adaptiveQuantization (p. 287)": enum,
        "interlaceMode (p. 287)": enum,
        "gopSize (p. 287)": number,
        "gopBReference (p. 287)": enum,
        "parDenominator (p. 287)": integer,
        "sceneChangeDetect (p. 287)": enum,
        "parControl (p. 287)": enum,
        "rateControlMode (p. 287)": enum
      },
      "proresSettings (p. 346)": {
        "framerateDenominator (p. 337)": integer,
        "slowPal (p. 337)": enum,
        "framerateControl (p. 337)": enum,
        "telecine (p. 337)": enum,
        "framerateConversionAlgorithm (p. 337)": enum,
        "interlaceMode (p. 337)": enum,
        "parNumerator (p. 337)": integer,
        "codecProfile (p. 337)": enum,
        "parDenominator (p. 338)": integer,
        "parControl (p. 338)": enum,
        "framerateNumerator (p. 338)": integer
      }
    }
  }
]}
"mpeg2Settings (p. 346)": {
  "minIInterval (p. 325)": integer,
  "parNumerator (p. 325)": integer,
  "gopSizeUnits (p. 325)": enum,
  "hrdBufferSize (p. 325)": integer,
  "qualityTuningLevel (p. 325)": enum,
  "maxBitrate (p. 325)": integer,
  "bitrate (p. 326)": integer,
  "spatialAdaptiveQuantization (p. 326)": enum,
  "slowPal (p. 326)": enum,
  "intraDcPrecision (p. 326)": enum,
  "codecProfile (p. 326)": enum,
  "softness (p. 326)": integer,
  "frameRateControl (p. 326)": enum,
  "telecine (p. 326)": enum,
  "codecLevel (p. 327)": enum,
  "framerateConversionAlgorithm (p. 327)": enum,
  "temporalAdaptiveQuantization (p. 327)": enum,
  "hrdBufferInitialFillPercentage (p. 327)": integer,
  "framerateDenominator (p. 327)": integer,
  "adaptiveQuantization (p. 328)": enum,
  "interlaceMode (p. 328)": enum,
  "gopSize (p. 328)": number,
  "syntax (p. 328)": enum,
  "parDenominator (p. 328)": integer,
  "sceneChangeDetect (p. 328)": enum,
  "parControl (p. 328)": enum,
  "rateControlMode (p. 329)": enum
},

"frameCaptureSettings (p. 346)": {
  "framerateDenominator (p. 270)": integer,
  "maxCaptures (p. 270)": integer,
  "framerateNumerators (p. 271)": integer,
  "quality (p. 271)": integer
},

"h264Settings (p. 346)": {
  "slices (p. 274)": integer,
  "minIInterval (p. 274)": integer,
  "parNumerator (p. 275)": integer,
  "flickerAdaptiveQuantization (p. 275)": enum,
  "gopSizeUnits (p. 275)": enum,
  "hrdBufferSize (p. 275)": integer,
  "qualityTuningLevel (p. 275)": enum,
  "maxBitrate (p. 275)": integer,
  "bitrate (p. 275)": integer,
  "spatialAdaptiveQuantization (p. 276)": enum,
  "slowPal (p. 276)": enum,
  "codecProfile (p. 276)": enum,
  "unregisteredSeiTimecode (p. 276)": enum,
  "softness (p. 276)": integer,
  "frameRateControl (p. 276)": enum,
  "telecine (p. 276)": enum,
  "codecLevel (p. 276)": enum,
  "framerateConversionAlgorithm (p. 276)": enum,
  "numberReferenceFrames (p. 277)": integer,
  "temporalAdaptiveQuantization (p. 277)": enum,
  "repeatPps (p. 277)": enum,
  "hrdBufferInitialFillPercentage (p. 277)": integer,
  "framerateDenominator (p. 277)": integer,
  "gopClosedCadence (p. 277)": integer,
  "numberBFramesBetweenReferenceFrames (p. 277)": integer,
  "framerateDenominator (p. 278)": integer,
  "entropyEncoding (p. 278)": enum,
"fieldEncoding (p. 278)" : enum,
"adaptiveQuantization (p. 278)" : enum,
"interlaceMode (p. 278)" : enum,
"gopSize (p. 278)" : number,
"gopBReference (p. 278)" : enum,
"syntax (p. 278)" : enum,
"parDenominator (p. 279)" : integer,
"sceneChangeDetect (p. 279)" : enum,
"parControl (p. 279)" : enum,
"rateControlMode (p. 279)" : enum
},
"afdSignaling (p. 347)" : enum,
"colorMetadata (p. 347)" : enum,
"timecodeInsertion (p. 347)" : enum,
"width (p. 347)" : integer,
"videoPreprocessors (p. 347)" : {
  "noiseReducer (p. 348)" : {
    "filter (p. 331)" : enum,
    "filterSettings (p. 331)" : {
      "strength (p. 332)" : integer
    },
    "spatialFilterSettings (p. 331)" : {
      "strength (p. 332)" : integer,
      "postFilterSharpenStrength (p. 332)" : integer,
      "speed (p. 332)" : integer
    }
  },
  "timecodeBurnin (p. 348)" : {
    "prefix (p. 342)" : "string",
    "fontSize (p. 342)" : integer,
    "position (p. 342)" : enum
  },
  "colorCorrector (p. 349)" : {
    "saturation (p. 252)" : integer,
    "brightness (p. 252)" : integer,
    "hdr10Metadata (p. 252)" : {
      "redPrimaryY (p. 289)" : integer,
      "greenPrimaryY (p. 289)" : integer,
      "whitePointX (p. 289)" : integer,
      "maxLuminance (p. 289)" : integer,
      "greenPrimaryX (p. 290)" : integer,
      "whitePointY (p. 290)" : integer,
      "redPrimaryX (p. 290)" : integer,
      "bluePrimaryX (p. 290)" : integer,
      "bluePrimaryY (p. 290)" : integer,
      "maxFrameAverageLightLevel (p. 291)" : integer,
      "maxContentLightLevel (p. 291)" : integer,
      "minLuminance (p. 291)" : integer
    },
    "contrast (p. 253)" : integer,
    "hue (p. 253)" : integer,
    "colorSpaceConversion (p. 253)" : enum
  },
  "deinterlacer (p. 349)" : {
    "mode (p. 257)" : enum,
    "control (p. 257)" : enum,
    "algorithm (p. 257)" : enum
  },
  "imageInserter (p. 349)" : {
    "insertableImages (p. 299)" : [
      {
        "duration (p. 302)" : integer,
        "fadeOut (p. 302)" : integer,
        "imageY (p. 303)" : integer,
        "fadeIn (p. 303)" : integer,
        "imageX (p. 303)" : integer
      }
    ]
  }
}
"imageX (p. 303)": integer,
"width (p. 303)": integer,
"startTime (p. 303)": "string",
"opacity (p. 304)": integer,
"layer (p. 304)": integer,
"height (p. 304)": integer,
"imageInserterInput (p. 304)": "string"
}
]
}
"antiAlias (p. 347)": enum,
"position (p. 347)": {
  "width (p. 338)": integer,
  "x (p. 339)": integer,
  "y (p. 339)": integer,
  "height (p. 339)": integer
},
"sharpness (p. 348)": integer,
"crop (p. 348)": {
  "width (p. 338)": integer,
  "x (p. 339)": integer,
  "y (p. 339)": integer,
  "height (p. 339)": integer
},
"dropFrameTimecode (p. 348)": enum,
"height (p. 348)": integer
},
"audioDescriptions (p. 333)": [
  {
    "audioTypeControl (p. 238)": enum,
    "languageCodeControl (p. 238)": enum,
    "remixSettings (p. 238)": {
      "channelsOut (p. 339)": integer,
      "channelMapping (p. 339)": {
        "outputChannels (p. 248)": [
          {
            "inputChannels (p. 334)": [integer
          ]
        ]
      }
    },
    "channelsIn (p. 339)": integer
  },
  "audioType (p. 238)": integer,
  "audioSourceName (p. 239)": "string",
  "codecSettings (p. 239)": {
    "codec (p. 237)": enum,
    "wavSettings (p. 237)": {
      "channels (p. 350)": integer,
      "bitDepth (p. 350)": integer,
      "format (p. 351)": enum,
      "sampleRate (p. 351)": integer
    },
    "aacSettings (p. 237)": {
      "vbrQuality (p. 233)": enum,
      "codecProfile (p. 233)": enum,
      "codingMode (p. 233)": enum,
      "specification (p. 233)": enum,
      "bitrate (p. 233)": integer,
      "rawFormat (p. 233)": enum,
      "rateControlMode (p. 233)": enum,
      "sampleRate (p. 233)": integer,
      "audioDescriptionBroadcasterMix (p. 234)": enum
    }
  }
],
"ac3Settings (p. 238)": {
  "dynamicRangeCompressionProfile (p. 235)”: enum,  
  "dialnorm (p. 235)”: integer,  
  "codingMode (p. 235)”: enum,  
  "metadataControl (p. 235)”: enum,  
  "bitrate (p. 235)”: integer,  
  "lfeFilter (p. 236)”: enum,  
  "bitstreamMode (p. 236)”: enum,  
  "sampleRate (p. 236)”: integer
},
"aiffSettings (p. 238)": {
  "channels (p. 236)”: integer,  
  "bitDepth (p. 236)”: integer,  
  "sampleRate (p. 236)”: integer
},
"eac3Settings (p. 238)": {
  "dialnorm (p. 265)”: integer,  
  "passsthroughControl (p. 265)”: enum,  
  "metadataControl (p. 265)”: enum,  
  "bitrate (p. 265)”: integer,  
  "dynamicRangeCompressionRf (p. 266)”: enum,  
  "sampleRate (p. 266)”: integer,  
  "surroundExMode (p. 266)”: enum,  
  "l1RtRtSurroundMixLevel (p. 266)”: number,  
  "dynamicRangeCompressionLine (p. 266)”: enum,  
  "lfeControl (p. 266)”: enum,  
  "codingMode (p. 266)”: enum,  
  "surroundMode (p. 266)”: enum,  
  "attenuationControl (p. 267)”: enum,  
  "lfeFilter (p. 267)”: enum,  
  "phaseControl (p. 267)”: enum,  
  "l1RtCenterMixLevel (p. 267)”: number,  
  "dcFilter (p. 267)”: enum,  
  "bitstreamMode (p. 267)”: enum,  
  "stereoDownmix (p. 267)”: enum,  
  "loRoSurroundMixLevel (p. 267)”: number,  
  "loRoCenterMixLevel (p. 267)”: number
},
"mp2Settings (p. 238)": {
  "channels (p. 321)”: integer,  
  "bitrate (p. 321)”: integer,  
  "sampleRate (p. 321)”: integer
},
"languageCode (p. 239)”: enum,
"streamName (p. 239)”: "string",
"audioNormalizationSettings (p. 239)": {
  "targetLkfs (p. 240)”: number,  
  "algorithmControl (p. 241)”: enum,  
  "loudnessLogging (p. 241)”: enum,  
  "peakCalculation (p. 241)”: enum,  
  "correctionGateLevel (p. 241)”: integer,  
  "algorithm (p. 241)”: enum
},
"customLanguageCode (p. 239)”: "string"
}],
"containerSettings (p. 333)": {
  "container (p. 254)”: enum,
  "mp4Settings (p. 254)": {
    "mp4MajorBrand (p. 322)”: "string",  
    "moovPlacement (p. 322)”: enum,  
    "csigAtom (p. 322)”: enum,  
    "freeSpaceBox (p. 322)”: enum
  },
  "m3u8Settings (p. 254)": {
    ""p. 254)": enum,
    "mp4Settings (p. 254)": {
      "mp4MajorBrand (p. 322)”: "string",  
      "moovPlacement (p. 322)”: enum,  
      "csigAtom (p. 322)”: enum,  
      "freeSpaceBox (p. 322)”: enum
    }
  }
}
"pmtPid (p. 317)": integer,
"nielsenId3 (p. 318)": enum,
"pcrPid (p. 318)": integer,
"audioPids (p. 318)": [
  integer
],
"audioFramesPerPes (p. 318)": integer,
"scte35Pid (p. 318)": integer,
"transportStreamId (p. 318)": integer,
"pcrControl (p. 318)": enum,
"videoPid (p. 319)": integer,
"privateMetadataPid (p. 319)": integer,
"pmtInterval (p. 319)": integer,
"patInterval (p. 319)": integer,
"programNumber (p. 319)": integer,
"timedMetadataPid (p. 319)": integer,
"timedMetadata (p. 320)": enum,
"scte35Source (p. 320)": enum
],
"m2tsSettings (p. 254)": {
  "dvbTeletextPid (p. 312)": integer,
  "nielsenId3 (p. 312)": enum,
  "bitrate (p. 312)": integer,
  "segmentationTime (p. 313)": number,
  "audioPids (p. 313)": [
    integer
  ],
  "rateMode (p. 313)": enum,
  "maxPcrInterval (p. 313)": integer,
  "audioFramesPerPes (p. 313)": integer,
  "ebpAudioInterval (p. 313)": enum,
  "fragmentTime (p. 313)": number,
  "scte35Pid (p. 314)": integer,
  "pmtInterval (p. 314)": integer,
  "pcrPid (p. 314)": integer,
  "privateMetadataPid (p. 314)": integer,
  "segmentationStyle (p. 314)": enum,
  "audioBufferModel (p. 314)": enum,
  "programNumber (p. 314)": integer,
  "dvbNitSettings (p. 314)": {
    "networkName (p. 258)": "string",
    "networkId (p. 258)": integer,
    "nitInterval (p. 258)": integer
  },
  "timedMetadataPid (p. 315)": integer,
  "scte35Source (p. 315)": enum,
  "pmtPid (p. 315)": integer,
  "bufferModel (p. 315)": integer,
  "ebpPlacement (p. 315)": enum,
  "dvbSdtSettings (p. 315)": {
    "sdtInterval (p. 258)": integer,
    "serviceName (p. 259)": "string",
    "serviceProviderName (p. 259)": "string",
    "outputSdt (p. 259)": enum
  },
  "nullPacketBitrate (p. 315)": number,
  "pcrPid (p. 315)": integer,
  "minEbpInterval (p. 316)": integer,
  "transportStreamId (p. 316)": integer,
  "videoPid (p. 316)": integer,
  "pcrControl (p. 316)": enum,
  "esRateInPes (p. 316)": enum,
  "segmentationMarkers (p. 316)": enum,
  "dvbTdtSettings (p. 316)": {
    "tdtInterval (p. 263)": integer
  },
  "patInterval (p. 317)": integer,
"dvbSubPids (p. 317)": [
    integer
  ],
"movSettings (p. 254)": {
  "reference (p. 321)": enum,
  "paddingControl (p. 321)": enum,
  "mpeg2FourCCControl (p. 321)": enum,
  "cs1gAtom (p. 321)": enum,
  "clapAtom (p. 321)": enum
},
"f4vSettings (p. 254)": {
  "moovPlacement (p. 269)": enum
},
"preset (p. 333)": "string",
"outputSettings (p. 333)": {
  "hlsSettings (p. 335)": {
    "iFrameOnlyManifest (p. 298)": enum,
    "audioRenditionSets (p. 298)": "string",
    "audioTrackType (p. 298)": enum,
    "segmentModifier (p. 298)": "string",
    "audioGroupId (p. 298)": "string"
  }
},
"captionDescriptions (p. 333)": [
  {
    "captionSelectorName (p. 245)": "string",
    "languageDescription (p. 246)": "string",
    "destinationSettings (p. 246)": {
      "burninDestinationSettings (p. 246)": {
        "xPosition (p. 242)": integer,
        "backgroundColor (p. 242)": enum,
        "teletextSpacing (p. 242)": enum,
        "yPosition (p. 242)": integer,
        "backgroundOpacity (p. 242)": integer,
        "fontOpacity (p. 243)": integer,
        "fontResolution (p. 243)": integer,
        "shadowOpacity (p. 243)": integer,
        "shadowXOffset (p. 244)": integer,
        "outlineSize (p. 243)": integer,
        "outlineColor (p. 243)": enum,
        "fontSize (p. 244)": integer,
        "shadowYOffset (p. 244)": integer,
        "alignment (p. 244)": enum,
        "shadowColor (p. 244)": enum,
        "fontColor (p. 244)": enum
      },
      "teletextDestinationSettings (p. 246)": {
        "pageNumber (p. 342)": "string"
      }
    },
    "ttmlDestinationSettings (p. 246)": {
      "stylePassthrough (p. 344)": enum
    },
    "destinationType (p. 246)": enum,
    "dvbSubDestinationSettings (p. 247)": {
      "xPosition (p. 259)": integer,
      "backgroundColor (p. 259)": enum,
      "teletextSpacing (p. 259)": enum,
      "yPosition (p. 259)": integer,
      "backgroundOpacity (p. 260)": integer,
      "fontOpacity (p. 260)": integer,
      "fontResolution (p. 260)": integer,
      "shadowOpacity (p. 260)": integer,
      "shadowXOffset (p. 260)": integer,
      "outlineSize (p. 261)": integer,
"outlineColor (p. 261)": enum,
"fontSize (p. 261)": integer,
"shadowXOffset (p. 261)": integer,
"alignment (p. 261)": enum,
"shadowColor (p. 261)": enum,
"fontColor (p. 261)": enum
},
"sccDestinationSettings (p. 247)": {
  "framerate (p. 340)": enum
}
},
"languageCode (p. 246)": enum,
"customLanguageCode (p. 246)": "string"
},
"nameModifier (p. 333)": "string"
],
"outputGroupSettings (p. 334)": {
  "dashIsoGroupSettings (p. 334)": {
    "fragmentLength (p. 255)": integer,
    "baseUrl (p. 255)": "string",
    "minBufferTime (p. 255)": integer,
    "encryption (p. 255)": {
      "spekeKeyProvider (p. 255)": {
        "resourceId (p. 340)": "string",
        "systemIds (p. 341)": [
          "string"
        ],
        "url (p. 341)": "string"
      }
    },
    "destination (p. 256)": "string",
    "segmentLength (p. 256)": integer,
    "segmentControl (p. 256)": enum,
    "hbbtvCompliance (p. 256)": enum
  },
  "fileGroupSettings (p. 334)": {
    "destination (p. 269)": "string"
  },
  "msSmoothGroupSettings (p. 335)": {
    "fragmentLength (p. 330)": integer,
    "encryption (p. 330)": {
      "spekeKeyProvider (p. 330)": {
        "resourceId (p. 340)": "string",
        "systemIds (p. 341)": [
          "string"
        ],
        "url (p. 341)": "string"
      }
    },
    "audioDeduplication (p. 330)": enum,
    "manifestEncoding (p. 330)": enum,
    "destination (p. 330)": "string"
  },
  "cmafGroupSettings (p. 335)": {
    "writeHlsManifest (p. 249)": enum,
    "writeDashManifest (p. 249)": enum,
    "fragmentLength (p. 249)": integer,
    "streamInfResolution (p. 250)": enum,
    "minBufferTime (p. 250)": integer,
    "clientCache (p. 250)": enum,
    "codecSpecification (p. 250)": enum,
    "destination (p. 250)": "string",
    "segmentControl (p. 250)": enum,
    "baseUrl (p. 250)": "string",
    208
"manifestDurationFormat (p. 296)"
  : enum
},
"name (p. 334)"
  : "string",
"customName (p. 334)"
  : "string"
],
"nielsenConfiguration (p. 306)"
  : {
"distributorId (p. 331)"
  : "string",
"breakoutCode (p. 331)"
  : integer
},
"inputs (p. 306)"
  : [
  {
"audioSelectors (p. 300)"
    : {
    },
"audioSelectorGroups (p. 301)"
    : {
    },
"filterEnable (p. 301)"
    : enum,
"deblockFilter (p. 301)"
    : enum,
"videoSelector (p. 301)"
    : {
"colorSpace (p. 349)"
      : enum,
"hdr10Metadata (p. 349)"
      : {
"redPrimaryY (p. 289)"
        : integer,
"greenPrimaryY (p. 289)"
        : integer,
"whitePointX (p. 289)"
        : integer,
"maxLuminance (p. 289)"
        : integer,
"greenPrimaryX (p. 290)"
        : integer,
"whitePointY (p. 290)"
        : integer,
"redPrimaryX (p. 290)"
        : integer,
"bluePrimaryX (p. 290)"
        : integer,
"bluePrimaryY (p. 290)"
        : integer,
"maxFrameAverageLightLevel (p. 291)"
        : integer,
"maxContentLightLevel (p. 291)"
        : integer,
"minLuminance (p. 291)"
        : integer
},
"programNumber (p. 349)"
  : integer,
"pid (p. 349)"
  : integer,
"colorSpaceUsage (p. 350)"
  : enum
},
"filterStrength (p. 301)"
  : integer,
"programNumber (p. 301)"
  : integer,
"timecodeSource (p. 301)"
  : enum,
"denoiseFilter (p. 301)"
  : enum,
"captionSelectors (p. 302)"
  : {
  },
"inputClippings (p. 302)"
  : [
  {
  "startTimecode (p. 299)"
    : "string",
  "endTimecode (p. 299)"
    : "string"
  }
],
"psiControl (p. 302)"
  : enum
],
"timedMetadataInsertion (p. 306)"
  : {
"id3Insertions (p. 344)"
    : [
    {
"id3 (p. 299)"
      : "string",
"timecode (p. 299)"
      : "string"
    }
  ],
"availBlanking (p. 306)"
  : {
"availBlankingImage (p. 241)"
    : "string"
  }
}
Example DELETE

```
{
  "name (p. 257)" : "string"
}
```

Response Bodies

Example GetJobTemplateResponse

```
{
  "jobTemplate (p. 271)" : {
    "createdAt (p. 304)" : "string",
    "lastUpdated (p. 304)" : "string",
    "settings (p. 305)" : {
      "timecodeConfig (p. 306)" : {
        "timestampOffset (p. 343)" : "string",
        "anchor (p. 343)" : "string",
        "start (p. 343)" : "string",
        "source (p. 344)" : enum
      },
      "adAvailOffset (p. 306)" : integer,
      "outputGroups (p. 306)" : [
        {
          "outputs (p. 334)" : [
            {
              "extension (p. 332)" : "string",
              "videoDescription (p. 333)" : {
                "fixedAfd (p. 346)" : integer,
                "scalingBehavior (p. 346)" : enum,
                "respondToAfd (p. 347)" : enum,
                "codecSettings (p. 347)" : {
                  "h265Settings (p. 346)" : {
                    "slices (p. 283)" : integer,
                    "minIInterval (p. 283)" : integer,
                    "parNumerator (p. 283)" : integer,
                    "flickerAdaptiveQuantization (p. 283)" : enum,
                    "gopSizeUnits (p. 283)" : enum,
                    "hrdBufferSize (p. 284)" : integer,
                    "qualityTuningLevel (p. 284)" : enum,
                    "maxBitrate (p. 284)" : integer,
                    "bitrate (p. 284)" : integer,
                    "spatialAdaptiveQuantization (p. 284)" : enum,
                    "sampleAdaptiveOffsetFilterMode (p. 284)" : enum,
                    "temporalIds (p. 284)" : enum,
                    "slowPal (p. 284)" : enum,
                    "tiles (p. 285)" : enum,
                    "writeMp4PackagingType (p. 285)" : enum,
                    "codecProfile (p. 285)" : enum,
                    "alternateTransferFunctionSei (p. 285)" : enum,
                    "unregisteredSeiTimecode (p. 285)" : enum,
                    "framerateControl (p. 285)" : enum,
                    "telecine (p. 285)" : enum,
                    "codecLevel (p. 285)" : enum,
                    "framerateConversionAlgorithm (p. 285)" : enum,
                    "numberReferenceFrames (p. 285)" : integer,
                }
              }
            }
          }
        }
      }
    }
  }
}
```
"temporalAdaptiveQuantization (p. 286)": enum,
"hrdBufferInitialFillPercentage (p. 286)": integer,
"framerateNumerator (p. 286)": integer,
"gopClosedCadence (p. 286)": integer,
"numberBFramesBetweenReferenceFrames (p. 286)": integer,
"framerateDenominator (p. 286)": integer,
"adaptiveQuantization (p. 287)": enum,
"interlaceMode (p. 287)": enum,
"gopSize (p. 287)": number,
"gopBReference (p. 287)": enum,
"parDenominator (p. 287)": integer,
"sceneChangeDetect (p. 287)": enum,
"parControl (p. 287)": enum,
"rateControlMode (p. 287)": enum
},
"codec (p. 346)": enum,
"proresSettings (p. 346)": {
  "framerateDenominator (p. 337)": integer,
  "slowPal (p. 337)": enum,
  "framerateControl (p. 337)": enum,
  "telecine (p. 337)": enum,
  "framerateConversionAlgorithm (p. 337)": enum,
  "interlaceMode (p. 337)": enum,
  "parNumerator (p. 337)": integer,
  "codecProfile (p. 337)": enum,
  "parControl (p. 338)": enum,
  "framerateNumerator (p. 338)": integer
},
"mpeg2Settings (p. 346)": {
  "minIInterval (p. 325)": integer,
  "parNumerator (p. 325)": integer,
  "gopSizeUnits (p. 325)": enum,
  "hrdBufferSize (p. 325)": integer,
  "qualityTuningLevel (p. 325)": enum,
  "maxBitrate (p. 325)": integer,
  "bitrate (p. 326)": integer,
  "spatialAdaptiveQuantization (p. 326)": enum,
  "slowPal (p. 326)": enum,
  "intraDcPrecision (p. 326)": enum,
  "codecProfile (p. 326)": enum,
  "softness (p. 326)": integer,
  "framerateControl (p. 326)": enum,
  "telecine (p. 326)": enum,
  "codecLevel (p. 327)": enum,
  "framerateConversionAlgorithm (p. 327)": enum,
  "temporalAdaptiveQuantization (p. 327)": enum,
  "hrdBufferInitialFillPercentage (p. 327)": integer,
  "framerateNumerator (p. 327)": integer,
  "gopClosedCadence (p. 327)": integer,
  "numberBFramesBetweenReferenceFrames (p. 327)": integer,
  "framerateDenominator (p. 328)": integer,
  "adaptiveQuantization (p. 328)": integer,
  "interlaceMode (p. 328)": enum,
  "gopSize (p. 328)": number,
  "syntax (p. 328)": enum,
  "parDenominator (p. 328)": integer,
  "sceneChangeDetect (p. 328)": enum,
  "parControl (p. 328)": enum,
  "rateControlMode (p. 329)": enum
},
"frameCaptureSettings (p. 346)": {
  "framerateDenominator (p. 270)": integer,
  "maxCaptures (p. 270)": integer,
  "framerateNumerator (p. 271)": integer,
  "quality (p. 271)": integer
}
"h264Settings (p. 346)”: {
  "slices (p. 274)”: integer,
  "minIInterval (p. 274)”: integer,
  "parNumerator (p. 275)”: integer,
  "flickerAdaptiveQuantization (p. 275)”: enum,
  "gopSizeUnits (p. 275)”: enum,
  "hrdBufferSize (p. 275)”: integer,
  "qualityTuningLevel (p. 275)”: enum,
  "maxBitrate (p. 275)”: integer,
  "bitrate (p. 275)”: integer,
  "spatialAdaptiveQuantization (p. 276)”: enum,
  "slowPal (p. 276)”: enum,
  "codecProfile (p. 276)”: enum,
  "unregisteredSeiTimecode (p. 276)”: enum,
  "softness (p. 276)”: integer,
  "framerateControl (p. 276)”: enum,
  "telecine (p. 276)”: enum,
  "codecLevel (p. 276)”: enum,
  "framerateConversionAlgorithm (p. 276)”: enum,
  "numberReferenceFrames (p. 277)”: integer,
  "temporalAdaptiveQuantization (p. 277)”: enum,
  "repeatPps (p. 277)”: enum,
  "hrdBufferInitialFillPercentage (p. 277)”: integer,
  "framerateNumerator (p. 278)”: integer,
  "gopClosedCadence (p. 278)”: integer,
  "numberBFramesBetweenReferenceFrames (p. 277)”: integer,
  "framerateDenominator (p. 278)”: integer,
  "entropyEncoding (p. 278)”: enum,
  "fieldEncoding (p. 278)”: enum,
  "adaptiveQuantization (p. 278)”: enum,
  "interlaceMode (p. 278)”: enum,
  "gopSize (p. 278)”: number,
  "gopBReference (p. 278)”: enum,
  "syntax (p. 278)”: enum,
  "parDenominator (p. 279)”: integer,
  "sceneChangeDetect (p. 279)”: integer,
  "parControl (p. 279)”: enum,
  "rateControlMode (p. 279)”: enum
}
"afdSignaling (p. 347)”: enum,
"colorMetadata (p. 347)”: enum,
"timecodeInsertion (p. 347)”: enum,
"width (p. 347)”: integer,
"videoPreprocessors (p. 347)”: {
  "noiseReducer (p. 348)”: {
    "filter (p. 331)”: enum,
    "filterSettings (p. 331)”: {
      "strength (p. 332)”: integer
    },
    "spatialFilterSettings (p. 331)”: {
      "strength (p. 332)”: integer,
      "postFilterSharpenStrength (p. 332)”: integer,
      "speed (p. 332)”: integer
    }
  },
  "timecodeBurnin (p. 348)”: {
    "prefix (p. 342)”: "string",
    "fontSize (p. 342)”: integer,
    "position (p. 342)”: enum
  },
  "colorCorrector (p. 349)”: {
    "saturation (p. 252)”: integer,
    "brightness (p. 252)”: integer,
    "hdr10Metadata (p. 252)”: {
}
"213"
"redPrimaryY (p. 289)": integer,
"greenPrimaryY (p. 289)": integer,
"whitePointX (p. 289)": integer,
"maxLuminance (p. 289)": integer,
"greenPrimaryX (p. 290)": integer,
"whitePointY (p. 290)": integer,
"redPrimaryX (p. 290)": integer,
"bluePrimaryX (p. 290)": integer,
"bluePrimaryY (p. 290)": integer,
"maxFrameAverageLightLevel (p. 291)": integer,
"maxContentLightLevel (p. 291)": integer,
"minLuminance (p. 291)": integer
},
"contrast (p. 253)": integer,
"hue (p. 253)": integer,
"colorSpaceConversion (p. 253)": enum
},
"deinterlacer (p. 349)": {
  "mode (p. 257)": enum,
  "control (p. 257)": enum,
  "algorithm (p. 257)": enum
},
"imageInserter (p. 349)": {
  "insertableImages (p. 299)": [
    {
      "duration (p. 302)": integer,
      "fadeOut (p. 302)": integer,
      "imageY (p. 303)": integer,
      "fadeIn (p. 303)": integer,
      "imageX (p. 303)": integer,
      "width (p. 303)": integer,
      "startTime (p. 303)": "string",
      "opacity (p. 304)": integer,
      "layer (p. 304)": integer,
      "height (p. 304)": integer,
      "imageInserterInput (p. 304)": "string"
    }
  ]
},
"antiAlias (p. 347)": enum,
"position (p. 347)": {
  "width (p. 338)": integer,
  "x (p. 339)": integer,
  "y (p. 339)": integer,
  "height (p. 339)": integer
},
"sharpness (p. 348)": integer,
"crop (p. 348)": {
  "width (p. 338)": integer,
  "x (p. 339)": integer,
  "y (p. 339)": integer,
  "height (p. 339)": integer
},
"dropFrameTimecode (p. 348)": enum,
"height (p. 348)": integer
},
"audioDescriptions (p. 333)": [
  {
    "audioTypeControl (p. 238)": enum,
    "languageCodeControl (p. 238)": enum,
    "remixSettings (p. 238)": {
      "channelsOut (p. 339)": integer,
      "channelMapping (p. 339)": {
        "outputChannels (p. 248)": [  
          
        ]
      }
    }
  }
]
"inputChannels (p. 334)" : [ integer ]
}
},
"channelsIn (p. 339)" : integer
},
"audioType (p. 238)" : integer,
"audioSourceName (p. 239)" : "string",
"codecSettings (p. 239)" : {
"codec (p. 237)" : enum,
"wavSettings (p. 237)" : {
"channels (p. 350)" : integer,
"bitDepth (p. 350)" : integer,
"format (p. 351)" : enum,
"sampleRate (p. 351)" : integer
},
"aacSettings (p. 237)" : {
"vbrQuality (p. 233)" : enum,
"codecProfile (p. 233)" : enum,
"codingMode (p. 233)" : enum,
"specification (p. 233)" : enum,
"bitrate (p. 233)" : integer,
"rawFormat (p. 233)" : enum,
"rateControlMode (p. 233)" : enum,
"sampleRate (p. 233)" : integer,
"audioDescriptionBroadcasterMix (p. 234)" : enum
},
"ac3Settings (p. 238)" : {
"dynamicRangeCompressionProfile (p. 235)" : enum,
"dialnorm (p. 235)" : integer,
"codingMode (p. 235)" : enum,
"metadataControl (p. 235)" : enum,
"bitrate (p. 235)" : integer,
"lifeFilter (p. 236)" : enum,
"bitstreamMode (p. 236)" : enum,
"sampleRate (p. 236)" : integer
},
"aiffSettings (p. 238)" : {
"channels (p. 236)" : integer,
"bitDepth (p. 236)" : integer,
"sampleRate (p. 236)" : integer
},
"eac3Settings (p. 238)" : {
"dialnorm (p. 265)" : integer,
"passthroughControl (p. 265)" : enum,
"metadataControl (p. 265)" : enum,
"bitrate (p. 265)" : integer,
"dynamicRangeCompressionRf (p. 266)" : enum,
"sampleRate (p. 266)" : integer,
"surroundExMode (p. 266)" : enum,
"ltRtSurroundMixLevel (p. 266)" : number,
"dynamicRangeCompressionLine (p. 266)" : enum,
"lifeControl (p. 266)" : enum,
"codingMode (p. 266)" : enum,
"surroundMode (p. 266)" : enum,
"attenuationControl (p. 267)" : enum,
"lifeFilter (p. 267)" : enum,
"phaseControl (p. 267)" : enum,
"ltRtCenterMixLevel (p. 267)" : number,
"dcFilter (p. 267)" : enum,
"bitstreamMode (p. 267)" : enum,
"stereoDownmix (p. 267)" : enum,
"loRoSurroundMixLevel (p. 267)" : number,
"loRoCenterMixLevel (p. 268)" : number
}
Schemas

"mp2Settings (p. 238)": {
  "channels (p. 321)": integer,
  "bitrate (p. 321)": integer,
  "sampleRate (p. 321)": integer
}

"languageCode (p. 239)": enum,
"streamName (p. 239)": "string",
"audioNormalizationSettings (p. 239)": {
  "targetLkfs (p. 240)": number,
  "algorithmControl (p. 241)": enum,
  "loudnessLogging (p. 241)": enum,
  "peakCalculation (p. 241)": enum,
  "correctionGateLevel (p. 241)": integer,
  "algorithm (p. 241)": enum
},
"customLanguageCode (p. 239)": "string"
}

"containerSettings (p. 333)": {
  "container (p. 254)": enum,
  "mp4Settings (p. 254)": {
    "mp4MajorBrand (p. 322)": "string",
    "moovPlacement (p. 322)": enum,
    "cslgAtom (p. 322)": enum,
    "freeSpaceBox (p. 322)": enum
  },
  "m3u8Settings (p. 254)": {
    "pmtPid (p. 317)": integer,
    "nielsenId3 (p. 318)": enum,
    "pcrPid (p. 318)": integer,
    "audioPids (p. 318)": [ integer
  ],
  "audioFramesPerPes (p. 318)": integer,
  "scte35Pid (p. 318)": integer,
  "transportStreamId (p. 318)": integer,
  "pcrControl (p. 318)": enum,
  "videoPid (p. 319)": integer,
  "privateMetadataPid (p. 319)": integer,
  "pmtInterval (p. 319)": integer,
  "patInterval (p. 319)": integer,
  "segmentationStyle (p. 314)": enum,
  "audioBufferModel (p. 314)": enum
},
"m2tsSettings (p. 254)": {
  "dvbTeletextPid (p. 312)": integer,
  "nielsenId3 (p. 312)": enum,
  "bitrate (p. 312)": integer,
  "segmentationTime (p. 313)": number,
  "audioPids (p. 313)": [ integer
  ],
  "rateMode (p. 313)": enum,
  "maxPcrInterval (p. 313)": integer,
  "audioFramesPerPes (p. 313)": integer,
  "ebpAudioInterval (p. 313)": enum,
  "fragmentTime (p. 313)": number,
  "scte35Pid (p. 314)": integer,
  "pmtInterval (p. 314)": integer,
  "privateMetadataPid (p. 314)": integer,
  "segmentationStyle (p. 314)": enum,
  "audioBufferModel (p. 314)": enum
}
"programNumber (p. 314)": integer,
"dvbNitSettings (p. 314)": {
   "networkName (p. 258)": "string",
   "networkId (p. 258)": integer,
   "nitInterval (p. 258)": integer
},
"timedMetadataPid (p. 315)": integer,
"scte35Source (p. 315)": enum,
"pmtPid (p. 315)": integer,
"bufferModel (p. 315)": enum,
"ebpPlacement (p. 315)": enum,
"dvbSdtSettings (p. 315)": {
   "sdtInterval (p. 258)": integer,
   "serviceName (p. 259)": "string",
   "serviceProviderName (p. 259)": "string",
   "outputSdt (p. 259)": "string",
   "nullPacketBitrate (p. 315)": number,
   "pcrPid (p. 315)": integer,
   "minEbpInterval (p. 316)": integer,
   "transportStreamId (p. 316)": integer,
   "videoPid (p. 316)": integer,
   "pcrControl (p. 316)": enum,
   "esRateInPes (p. 316)": integer,
   "segmentationMarkers (p. 316)": enum,
   "dvbTdtSettings (p. 316)": {
      "tdtInterval (p. 263)": integer
   },
   "patInterval (p. 317)": integer,
   "dvbSubPids (p. 317)": [ integer
   ]
},
"movSettings (p. 254)": {
   "reference (p. 321)": enum,
   "paddingControl (p. 321)": enum,
   "mpeg2FourCCControl (p. 321)": enum,
   "cslgAtom (p. 321)": enum,
   "clapAtom (p. 321)": enum
},
"f4vSettings (p. 254)": {
   "moovPlacement (p. 269)": enum
},
"preset (p. 333)": "string",
"outputSettings (p. 333)": {
   "hisSettings (p. 335)": {
      "iFrameOnlyManifest (p. 298)": "string",
      "audioRenditionSets (p. 298)": "string",
      "audioTrackType (p. 298)": "string",
      "segmentModifier (p. 298)": "string",
      "audioGroupId (p. 298)": "string"
   }
},
"captionDescriptions (p. 333)": [
   {
      "captionSelectorName (p. 245)": "string",
      "languageDescription (p. 246)": "string",
      "destinationSettings (p. 246)": {
         "burninDestinationSettings (p. 246)": {
            "xPosition (p. 242)": integer,
            "backgroundColor (p. 242)": integer,
            "teletextSpacing (p. 242)": integer,
            "yPosition (p. 242)": integer,
            "backgroundOpacity (p. 242)": integer,
            "fontOpacity (p. 243)": integer,
            "backgroundOpacity (p. 242)": integer,
        },
"fontResolution (p. 243)": integer,
"shadowOpacity (p. 243)": integer,
"shadowYOffset (p. 243)": integer,
"outlineSize (p. 243)": integer,
"outlineColor (p. 243)": enum,
"fontSize (p. 244)": integer,
"shadowXOffset (p. 244)": integer,
"alignment (p. 244)": enum,
"shadowColor (p. 244)": enum,
"fontColor (p. 244)": enum
],
"teletextDestinationSettings (p. 246)": {
"pageNumber (p. 342)": "string"
},
"ttmlDestinationSettings (p. 246)": {
"stylePassthrough (p. 344)": enum
},
"destinationType (p. 246)": enum,
"dvbSubDestinationSettings (p. 247)": {
"xPosition (p. 259)": integer,
"backgroundColor (p. 259)": enum,
"teletextSpacing (p. 259)": enum,
"yPosition (p. 259)": integer,
"backgroundOpacity (p. 260)": integer,
"fontOpacity (p. 260)": integer,
"fontResolution (p. 260)": integer,
"shadowOpacity (p. 260)": integer,
"shadowXOffset (p. 260)": integer,
"outlineSize (p. 261)": integer,
"outlineColor (p. 261)": enum,
"fontSize (p. 261)": integer,
"shadowColor (p. 261)": enum,
"fontColor (p. 261)": enum
},
"sccDestinationSettings (p. 247)": {
"framerate (p. 340)": enum
},
"languageCode (p. 246)": enum,
"customLanguageCode (p. 246)": "string"
]},
"nameModifier (p. 333)": "string"
]},
"outputGroupSettings (p. 334)": {
"dashIsoGroupSettings (p. 334)": {
"fragmentLength (p. 255)": integer,
"baseUrl (p. 255)": "string",
"minBufferTime (p. 255)": integer,
"encryption (p. 255)": {
"spekeKeyProvider (p. 255)": {
"resourceId (p. 340)": "string",
"systemIds (p. 341)": ["string"
],
"url (p. 341)": "string"
},
"destination (p. 256)": "string",
"segmentLength (p. 256)": integer,
"segmentControl (p. 256)": enum,
"hbbtvCompliance (p. 256)": enum
},
"dashIsoGroupSettings (p. 334)": {
"fragmentLength (p. 255)": integer,
"baseUrl (p. 255)": "string",
"minBufferTime (p. 255)": integer,
"encryption (p. 255)": {
"spekeKeyProvider (p. 255)": {
"resourceId (p. 340)": "string",
"systemIds (p. 341)": ["string"
],
"url (p. 341)": "string"
},
"destination (p. 256)": "string",
"segmentLength (p. 256)": integer,
"segmentControl (p. 256)": enum,
"hbbtvCompliance (p. 256)": enum
},
"dashIsoGroupSettings (p. 334)": {
"fragmentLength (p. 255)": integer,
"baseUrl (p. 255)": "string",
"minBufferTime (p. 255)": integer,
"encryption (p. 255)": {
"spekeKeyProvider (p. 255)": {
"resourceId (p. 340)": "string",
"systemIds (p. 341)": ["string"
],
"url (p. 341)": "string"
},
"destination (p. 256)": "string",
"segmentLength (p. 256)": integer,
"segmentControl (p. 256)": enum,
"hbbtvCompliance (p. 256)": enum
},
"dashIsoGroupSettings (p. 334)": {
"fragmentLength (p. 255)": integer,
"baseUrl (p. 255)": "string",
"minBufferTime (p. 255)": integer,
"encryption (p. 255)": {
"spekeKeyProvider (p. 255)": {
"resourceId (p. 340)": "string",
"systemIds (p. 341)": ["string"
],
"url (p. 341)": "string"
},
"destination (p. 256)": "string",
"segmentLength (p. 256)": integer,
"segmentControl (p. 256)": enum,
"hbbtvCompliance (p. 256)": enum
},
"dashIsoGroupSettings (p. 334)": {
"fragmentLength (p. 255)": integer,
"baseUrl (p. 255)": "string",
"minBufferTime (p. 255)": integer,
"encryption (p. 255)": {
"spekeKeyProvider (p. 255)": {
"resourceId (p. 340)": "string",
"systemIds (p. 341)": ["string"
],
"url (p. 341)": "string"
},
"destination (p. 256)": "string",
"segmentLength (p. 256)": integer,
"segmentControl (p. 256)": enum,
"hbbtvCompliance (p. 256)": enum
},
"dashIsoGroupSettings (p. 334)": {
"fragmentLength (p. 255)": integer,
"baseUrl (p. 255)": "string",
"minBufferTime (p. 255)": integer,
"encryption (p. 255)": {
"spekeKeyProvider (p. 255)": {
"resourceId (p. 340)": "string",
"systemIds (p. 341)": ["string"
],
"url (p. 341)": "string"
},
"destination (p. 256)": "string",
"segmentLength (p. 256)": integer,
"segmentControl (p. 256)": enum,
"hbbtvCompliance (p. 256)": enum
}
"fileGroupSettings (p. 334)": {
  "destination (p. 269)": "string"
},
"msSmoothGroupSettings (p. 335)": {
  "fragmentLength (p. 330)": integer,
  "encryption (p. 330)": {
    "spekeKeyProvider (p. 330)": {
      "resourceId (p. 340)": "string",
      "systemIds (p. 341)": [
        "string"
      ],
      "url (p. 341)": "string"
    },
    "url (p. 341)": "string"
  },
  "audioDeduplication (p. 330)": enum,
  "manifestEncoding (p. 330)": enum,
  "destination (p. 330)": "string"
},
"cmafGroupSettings (p. 335)": {
  "writeHlsManifest (p. 249)": enum,
  "writeDashManifest (p. 249)": enum,
  "fragmentLength (p. 249)": integer,
  "streamInfResolution (p. 250)": enum,
  "minBufferTime (p. 250)": integer,
  "clientCache (p. 250)": enum,
  "codecSpecification (p. 250)": enum,
  "destination (p. 250)": "string",
  "segmentControl (p. 250)": enum,
  "baseUrl (p. 250)": "string",
  "encryption (p. 250)": {
    "initializationVectorInManifest (p. 248)": enum,
    "constantInitializationVector (p. 249)": "string",
    "spekeKeyProvider (p. 249)": {
      "keyFormatVersions (p. 341)": "string",
      "keyFormat (p. 341)": "string",
      "staticKeyValue (p. 341)": "string",
      "url (p. 341)": "string"
    },
    "type (p. 249)": enum,
    "encryptionMethod (p. 249)": enum
  },
  "manifestCompress (p. 251)": enum,
  "segmentLength (p. 251)": integer,
  "manifestDurationFormat (p. 251)": enum
},
"type (p. 335)": enum,
"hlsGroupSettings (p. 335)": {
  "segmentsPerSubdirectory (p. 294)": integer,
  "streamInfResolution (p. 294)": enum,
  "timestampDeltaMilliseconds (p. 294)": integer,
  "outputSelection (p. 294)": enum,
  "captionLanguageMappings (p. 294)": [
    {
      "languageDescription (p. 292)": "string",
      "captionChannel (p. 292)": integer,
      "languageCode (p. 292)": enum,
      "customLanguageCode (p. 292)": "string"
    }
  ],
  "clientCache (p. 294)": enum,
  "codecSpecification (p. 295)": enum,
  "destination (p. 295)": "string",
  "timedMetadataId3Frame (p. 295)": enum,
  "segmentControl (p. 295)": enum,
  "timedMetadataId3Period (p. 295)": integer,
  "captionLanguageSetting (p. 295)": enum,
"minSegmentLength (p. 295)”: integer,
"directoryStructure (p. 295)”: enum,
"programDateTime (p. 296)”: enum,
"baseUrl (p. 296)”: “string”,
"encryption (p. 296)”: {
  "initializationVectorInManifest (p. 293)”: enum,
  "constantInitializationVector (p. 293)”: “string”,
  "staticKeyProvider (p. 293)”: {
    "keyFormatVersions (p. 341)”: “string”,
    "keyFormat (p. 341)”: “string”,
    "staticKeyValue (p. 341)”: “string”,
    "url (p. 341)”: “string”
  },
  "type (p. 293)”: enum,
  "encryptionMethod (p. 293)”: enum,
  "spekeKeyProvider (p. 293)”: {
    "resourceId (p. 340)”: “string”,
    "systemIds (p. 341)”: [
      "string"
    ],
    "url (p. 341)”: “string”
  },
  "adMarkers (p. 296)”: [
    enum
  ],
  "programDateTimePeriod (p. 296)”: integer,
  "manifestCompression (p. 296)”: enum,
  "segmentLength (p. 296)”: integer,
  "manifestDurationFormat (p. 296)”: enum
},
"name (p. 334)”: “string”,
"customName (p. 334)”: “string”
}],
"nielsenConfiguration (p. 306)”: {
  "distributorId (p. 331)”: “string”,
  "breakoutCode (p. 331)”: integer
},
"inputs (p. 306)”: [220
  {
    "audioSelectors (p. 300)”: {
    "audioSelectorGroups (p. 301)”: {
    },
    "filterEnable (p. 301)”: enum,
    "deblockFilter (p. 301)”: enum,
    "videoSelector (p. 301)”: {
      "colorSpace (p. 349)”: enum,
      "hdr10Metadata (p. 349)”: {
        "redPrimaryY (p. 289)”: integer,
        "greenPrimaryY (p. 289)”: integer,
        "whitePointX (p. 289)”: integer,
        "maxLuminance (p. 289)”: integer,
        "greenPrimaryX (p. 290)”: integer,
        "whitePointY (p. 290)”: integer,
        "redPrimaryX (p. 290)”: integer,
        "bluePrimaryX (p. 290)”: integer,
        "bluePrimaryY (p. 290)”: integer,
        "maxFrameAverageLightLevel (p. 291)”: integer,
        "maxContentLightLevel (p. 291)”: integer,
        "minLuminance (p. 291)": integer
      },
      "programNumber (p. 349)”: integer,
      "pid (p. 349)”: integer,
Example UpdateJobTemplateResponse

{
  "jobTemplate (p. 345)": {
    "createdAt (p. 304)": "string",
    "lastUpdated (p. 304)": "string",
    "settings (p. 305)": {
      "timecodeConfig (p. 306)": {
        "timestampOffset (p. 343)": "string",
        "anchor (p. 343)": "string",
        "start (p. 343)": "string",
        "source (p. 344)": enum
      },
      "adAvailOffset (p. 306)": integer,
      "outputGroups (p. 306)": [
        {
          "outputs (p. 334)": [
            {
              "extension (p. 332)": "string",
              "videoDescription (p. 333)": {
                "fixedAfd (p. 346)": integer,
                "scalingBehavior (p. 346)": enum,
                "respondToAfd (p. 347)": enum,
                "h265Settings (p. 346)": {
                  "slices (p. 283)": integer,
                  "colorSpaceUsage (p. 350)": enum,
                  "filterStrength (p. 301)": integer,
                  "programNumber (p. 301)": integer,
                  "timecodeSource (p. 301)": enum,
                  "denoiseFilter (p. 301)": enum,
                  "captionSelectors (p. 302)": {
                    "inputClippings (p. 302)": [
                      {
                        "startTimecode (p. 299)": "string",
                        "endTimecode (p. 299)": "string"
                      }
                    ],
                    "psiControl (p. 302)": enum
                  },
                  "timedMetadataInsertion (p. 306)": {
                    "id3Insertions (p. 344)": [
                      {
                        "id3 (p. 299)": "string",
                        "timecode (p. 299)": "string"
                      }
                    ],
                    "availBlanking (p. 306)": {
                      "availBlankingImage (p. 241)": "string"
                    },
                    "name (p. 305)": "string",
                    "description (p. 305)": "string",
                    "arn (p. 305)": "string",
                    "category (p. 305)": "string",
                    "type (p. 305)": enum,
                    "queue (p. 305)": "string"
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
"minIInterval (p. 283)" : integer,
"parNumerator (p. 283)" : integer,
"flickerAdaptiveQuantization (p. 283)" : enum,
"gopSizeUnits (p. 283)" : enum,
"hrdBufferSize (p. 284)" : integer,
"qualityTuningLevel (p. 284)" : enum,
"maxBitrate (p. 284)" : integer,
"bitrate (p. 284)" : integer,
"spatialAdaptiveQuantization (p. 284)" : enum,
"sampleAdaptiveOffsetFilterMode (p. 284)" : enum,
"temporalIds (p. 284)" : enum,
"slowPal (p. 284)" : enum,
"tiles (p. 285)" : enum,
"writeMp4PackagingType (p. 285)" : enum,
"codecProfile (p. 285)" : enum,
"alternateTransferFunctionSei (p. 285)" : enum,
"unregisteredSeiTimecode (p. 285)" : enum,
"framerateControl (p. 285)" : enum,
"telecine (p. 285)" : enum,
"codecLevel (p. 285)" : enum,
"framerateConversionAlgorithm (p. 285)" : enum,
"numberReferenceFrames (p. 285)" : integer,
"temporalAdaptiveQuantization (p. 286)" : enum,
"hrdBufferInitialFillPercentage (p. 286)" : integer,
"framerateNumerator (p. 286)" : integer,
"framerateDenominator (p. 286)" : integer,
"numberBFramesBetweenReferenceFrames (p. 286)" : integer,
"framerateConverter (p. 286)" : integer,
"adaptiveQuantization (p. 287)" : enum,
"interlaceMode (p. 287)" : enum,
"gopSize (p. 287)" : number,
"gopClosedCadence (p. 287)" : integer,
"parNumerator (p. 325)" : integer,
"parDenominator (p. 325)" : integer,
"parControl (p. 325)" : enum,
"framerateControl (p. 325)" : enum,
"framerateConversionAlgorithm (p. 325)" : enum,
"telecine (p. 325)" : enum,
"telecine (p. 326)" : enum,
"intraDcPrecision (p. 326)" : enum,
"framerateControl (p. 326)" : enum,
"codecLevel (p. 327)" : enum,
"framerateConversionAlgorithm (p. 327)" : enum,
"temporalAdaptiveQuantization (p. 327)" : enum,
"hrdBufferInitialFillPercentage (p. 327)" : integer,
"framerateNumerator (p. 327)" : integer,
"gopClosedCadence (p. 327)" : integer,
"numberOfFramesBetweenReferenceFrames (p. 327)" : integer,
"framerateDenominator (p. 328)" : integer,
"adaptiveQuantization (p. 328)" : enum,
"interlaceMode (p. 328)" : enum,
"gopSize (p. 328)" : number,
"syntax (p. 328)" : enum,
"parDenominator (p. 328)" : integer,
"sceneChangeDetect (p. 328)" : enum,
"parControl (p. 328)" : enum,
"rateControlMode (p. 329)" : enum
},
"frameCaptureSettings (p. 346)" : {
  "framerateDenominator (p. 270)" : integer,
  "maxCaptures (p. 270)" : integer,
  "framerateNumerator (p. 271)" : integer,
  "quality (p. 271)" : integer
},
"h264Settings (p. 346)" : {
  "slices (p. 274)" : integer,
  "minIInterval (p. 274)" : integer,
  "parNumerator (p. 275)" : integer,
  "flickerAdaptiveQuantization (p. 275)" : enum,
  "gopSizeUnits (p. 275)" : enum,
  "hrdBufferSize (p. 275)" : integer,
  "qualityTuningLevel (p. 275)" : enum,
  "maxBitrate (p. 275)" : integer,
  "bitrate (p. 275)" : integer,
  "spatialAdaptiveQuantization (p. 276)" : enum,
  "slowPal (p. 276)" : enum,
  "codecProfile (p. 276)" : enum,
  "unregisteredSeiTimecode (p. 276)" : enum,
  "softness (p. 276)" : integer,
  "framerateControl (p. 276)" : enum,
  "telecine (p. 276)" : enum,
  "codecLevel (p. 276)" : enum,
  "framerateConversionAlgorithm (p. 276)" : enum,
  "numberOfReferenceFrames (p. 277)" : integer,
  "temporalAdaptiveQuantization (p. 277)" : enum,
  "repeatPps (p. 277)" : enum,
  "hrdBufferInitialFillPercentage (p. 277)" : integer,
  "framerateNumerator (p. 277)" : integer,
  "gopClosedCadence (p. 277)" : integer,
  "numberOfFramesBetweenReferenceFrames (p. 277)" : integer,
  "framerateDenominator (p. 278)" : integer,
  "entropyEncoding (p. 278)" : enum,
  "fieldEncoding (p. 278)" : enum,
  "adaptiveQuantization (p. 278)" : enum,
  "interlaceMode (p. 278)" : enum,
  "gopSize (p. 278)" : number,
  "gopBReference (p. 278)" : enum,
  "syntax (p. 278)" : enum,
  "parDenominator (p. 279)" : integer,
  "sceneChangeDetect (p. 279)" : integer,
  "parControl (p. 279)" : enum,
  "rateControlMode (p. 279)" : enum
}
"width (p. 347)": integer,
"videoPreprocessors (p. 347)": {
    "noiseReducer (p. 348)": {
        "filter (p. 331)": enum,
        "filterSettings (p. 331)": {
            "strength (p. 332)": integer
        },
        "spatialFilterSettings (p. 331)": {
            "strength (p. 332)": integer,
            "postFilterSharpenStrength (p. 332)": integer,
            "speed (p. 332)": integer
        }
    },
    "timecodeBurnin (p. 348)": {
        "prefix (p. 342)": "string",
        "fontSize (p. 342)": integer,
        "position (p. 342)": enum
    },
    "colorCorrector (p. 349)": {
        "saturation (p. 252)": integer,
        "brightness (p. 252)": integer,
        "hdr10Metadata (p. 252)": {
            "redPrimaryY (p. 289)": integer,
            "greenPrimaryY (p. 289)": integer,
            "whitePointX (p. 289)": integer,
            "maxLuminance (p. 289)": integer,
            "greenPrimaryX (p. 290)": integer,
            "whitePointY (p. 290)": integer,
            "redPrimaryX (p. 290)": integer,
            "bluePrimaryX (p. 290)": integer,
            "bluePrimaryY (p. 290)": integer,
            "maxFrameAverageLightLevel (p. 291)": integer,
            "maxContentLightLevel (p. 291)": integer,
            "minLuminance (p. 291)": integer
        },
        "contrast (p. 253)": integer,
        "hue (p. 253)": integer,
        "colorSpaceConversion (p. 253)": enum
    },
    "deinterlacer (p. 349)": {
        "mode (p. 257)": enum,
        "control (p. 257)": enum,
        "algorithm (p. 257)": enum
    },
    "imageInserter (p. 349)": {
        "insertableImages (p. 299)": [
            {
                "duration (p. 302)": integer,
                "fadeOut (p. 302)": integer,
                "imageY (p. 303)": integer,
                "fadeIn (p. 303)": integer,
                "imageX (p. 303)": integer,
                "width (p. 303)": integer,
                "startTime (p. 303)": "string",
                "opacity (p. 304)": integer,
                "layer (p. 304)": integer,
                "height (p. 304)": integer,
                "imageInserterInput (p. 304)": "string"
            }
        ]
    }
},
"antiAlias (p. 347)": enum,
"position (p. 347)": {
    "width (p. 338)": integer,
    "x (p. 339)": integer,
"y (p. 339)" : integer,
"height (p. 339)" : integer
},
"sharpness (p. 348)" : integer,
"crop (p. 348)" : {
  "width (p. 338)" : integer,
  "x (p. 339)" : integer,
  "y (p. 339)" : integer,
  "height (p. 339)" : integer
},
"dropFrameTimecode (p. 348)" : enum,
"height (p. 348)" : integer
},
"audioDescriptions (p. 333)" : [
  {
    "audioTypeControl (p. 238)" : enum,
    "languageCodeControl (p. 238)" : enum,
    "remixSettings (p. 238)" : {
      "channelsOut (p. 339)" : integer,
      "channelMapping (p. 339)" : {
        "outputChannels (p. 248)" : [
          {
            "inputChannels (p. 334)" : [
              integer
            ]
          }
        ]
      },
      "channelsIn (p. 339)" : integer
    },
    "audioType (p. 238)" : integer,
    "audioSourceName (p. 239)" : "string",
    "codecSettings (p. 239)" : {
      "codec (p. 237)" : enum,
      "wavSettings (p. 237)" : {
        "channels (p. 350)" : integer,
        "bitDepth (p. 350)" : integer,
        "format (p. 351)" : enum,
        "sampleRate (p. 351)" : integer
      },
      "aacSettings (p. 237)" : {
        "vbrQuality (p. 233)" : enum,
        "codecProfile (p. 233)" : enum,
        "codingMode (p. 233)" : enum,
        "specification (p. 233)" : enum,
        "bitrate (p. 233)" : integer,
        "rawFormat (p. 233)" : enum,
        "rateControlMode (p. 233)" : enum,
        "sampleRate (p. 233)" : integer,
        "audioDescriptionBroadcasterMix (p. 234)" : enum
      },
      "ac3Settings (p. 238)" : {
        "dynamicRangeCompressionProfile (p. 235)" : enum,
        "dialnorm (p. 235)" : integer,
        "codingMode (p. 235)" : enum,
        "metaDataControl (p. 235)" : enum,
        "bitrate (p. 235)" : integer,
        "lifeFilter (p. 236)" : enum,
        "bitstreamMode (p. 236)" : enum,
        "sampleRate (p. 236)" : integer
      },
      "aiffSettings (p. 238)" : {
        "channels (p. 236)" : integer,
        "bitDepth (p. 236)" : integer,
        "sampleRate (p. 236)" : integer
      }
    }
  }
],
",225
"eac3Settings (p. 238)": {
    "dailnorm (p. 265)": integer,
    "passThroughControl (p. 265)": enum,
    "metadataControl (p. 265)": enum,
    "bitrate (p. 265)": integer,
    "dynamicRangeCompressionRf (p. 266)": enum,
    "sampleRate (p. 266)": integer,
    "surroundExMode (p. 266)": enum,
    "ltRtSurroundMixLevel (p. 266)": number,
    "dynamicRangeCompressionLine (p. 266)": enum,
    "lfeControl (p. 266)": enum,
    "codingMode (p. 266)": enum,
    "surroundMode (p. 266)": enum,
    "attenuationControl (p. 267)": enum,
    "lfeFilter (p. 267)": enum,
    "phaseControl (p. 267)": enum,
    "ltRtCenterMixLevel (p. 267)": number,
    "dcFilter (p. 267)": enum,
    "bitstreamMode (p. 267)": enum,
    "stereoDownmix (p. 267)": enum,
    "loRoSurroundMixLevel (p. 267)": number,
    "loRoCenterMixLevel (p. 268)": number
},
"mp2Settings (p. 238)": {
    "channels (p. 321)": integer,
    "bitrate (p. 321)": integer,
    "sampleRate (p. 321)": integer
},
"languageCode (p. 239)": enum,
"streamName (p. 239)": "string",
"audioNormalizationSettings (p. 239)": {
    "targetLkfs (p. 240)": number,
    "algorithmControl (p. 241)": enum,
    "loudnessLogging (p. 241)": enum,
    "peakCalculation (p. 241)": enum,
    "correctionGateLevel (p. 241)": integer,
    "algorithm (p. 241)": enum
},
"customLanguageCode (p. 239)": "string"
}
"containerSettings (p. 333)": {
    "container (p. 254)": enum,
    "mp4Settings (p. 254)": {
        "mp4MajorBrand (p. 322)": "string",
        "moovPlacement (p. 322)": enum,
        "csigAtom (p. 322)": enum,
        "freeSpaceBox (p. 322)": enum
    },
    "m3u8Settings (p. 254)": {
        "pmtPid (p. 317)": integer,
        "nielsenId3 (p. 318)": enum,
        "pcrPid (p. 318)": integer,
        "audioPids (p. 318)": [integer
    ],
    "audioFramesPerPes (p. 318)": integer,
    "scte35Pid (p. 318)": integer,
    "transportStreamId (p. 318)": integer,
    "pcrControl (p. 318)": enum,
    "videoPid (p. 319)": integer,
    "privateMetaDataPid (p. 319)": integer,
    "pmtInterval (p. 319)": integer,
    "patInterval (p. 319)": integer,
    "programNumber (p. 319)": integer,
"timedMetadataPid (p. 319)": integer,
"timedMetadata (p. 320)": enum,
"scte35Source (p. 320)": enum
},
"m2tsSettings (p. 254)": {
  "dvbTeletextPid (p. 312)": integer,
  "nielsenId3 (p. 312)": enum,
  "bitrate (p. 312)": integer,
  "segmentationTime (p. 313)": number,
  "audioPids (p. 313)": [
    integer
  ],
  "rateMode (p. 313)": enum,
  "maxPcrInterval (p. 313)": integer,
  "audioFramesPerPes (p. 313)": integer,
  "ebpAudioInterval (p. 313)": enum,
  "fragmentTime (p. 313)": number,
  "scte35Pid (p. 314)": integer,
  "pmtInterval (p. 314)": integer,
  "privateMetadataPid (p. 314)": integer,
  "segmentationStyle (p. 314)": enum,
  "audioBufferModel (p. 314)": enum,
  "programNumber (p. 314)": integer,
  "dvbNitSettings (p. 314)": {
    "networkName (p. 258)": "string",
    "networkId (p. 258)": integer,
    "nitInterval (p. 258)": integer
  },
  "timedMetadataPid (p. 315)": integer,
  "scte35Source (p. 315)": enum,
  "pmtPid (p. 315)": integer,
  "bufferModel (p. 315)": enum,
  "ebpPlacement (p. 315)": enum,
  "dvbSdtSettings (p. 315)": {
    "sdtInterval (p. 258)": integer,
    "serviceName (p. 259)": "string",
    "serviceProviderName (p. 259)": "string",
    "outputSdt (p. 259)": enum
  },
  "nullPacketBitrate (p. 315)": number,
  "pcrPid (p. 315)": integer,
  "minEbpInterval (p. 316)": integer,
  "transportStreamId (p. 316)": integer,
  "videoPid (p. 316)": integer,
  "pcrControl (p. 316)": enum,
  "esRateInPes (p. 316)": enum,
  "segmentationMarkers (p. 316)": enum,
  "dvbTdtSettings (p. 316)": {
    "tdtInterval (p. 263)": integer
  },
  "patInterval (p. 317)": integer,
  "dvbSubPids (p. 317)": [
    integer
  ],
  "movSettings (p. 254)": {
    "reference (p. 321)": enum,
    "paddingControl (p. 321)": enum,
    "mpeg2FourCCControl (p. 321)": enum,
    "cslgAtom (p. 321)": enum,
    "clapAtom (p. 321)": enum
  },
  "f4vSettings (p. 254)": {
    "moovPlacement (p. 269)": enum
  }
},
"nullPacketBitrate (p. 315)": number,
"preset (p. 333)": "string",
"outputSettings (p. 333)": {
  "hlsSettings (p. 335)": {
    "iFrameOnlyManifest (p. 298)": enum,
    "audioRenditionSets (p. 298)": "string",
    "audioTrackType (p. 298)": enum,
    "segmentModifier (p. 298)": "string",
    "audioGroupId (p. 298)": "string"
  },
  "captionDescriptions (p. 333)": [
    {
      "captionSelectorName (p. 245)": "string",
      "languageDescription (p. 246)": "string",
      "destinationSettings (p. 246)": {
        "burninDestinationSettings (p. 246)": {
          "xPosition (p. 242)": integer,
          "backgroundColor (p. 242)": enum,
          "teletextSpacing (p. 242)": enum,
          "yPosition (p. 242)": integer,
          "backgroundOpacity (p. 242)": integer,
          "fontOpacity (p. 243)": integer,
          "fontResolution (p. 243)": integer,
          "shadowOpacity (p. 243)": integer,
          "shadowYOffset (p. 243)": integer,
          "fontSize (p. 244)": integer,
          "shadowXOffset (p. 244)": integer,
          "alignment (p. 244)": enum,
          "shadowColor (p. 244)": enum,
          "fontName (p. 244)": enum
        },
        "teletextDestinationSettings (p. 246)": {
          "pageNumber (p. 342)": "string"
        },
        "ttmlDestinationSettings (p. 246)": {
          "stylePassthrough (p. 344)": enum
        },
        "destinationType (p. 246)": enum,
        "dvbSubDestinationSettings (p. 247)": {
          "xPosition (p. 259)": integer,
          "backgroundColor (p. 259)": enum,
          "teletextSpacing (p. 259)": enum,
          "yPosition (p. 259)": integer,
          "backgroundOpacity (p. 260)": integer,
          "fontOpacity (p. 260)": integer,
          "fontResolution (p. 260)": integer,
          "shadowOpacity (p. 260)": integer,
          "shadowXOffset (p. 260)": integer,
          "outlineSize (p. 261)": integer,
          "outlineColor (p. 261)": enum,
          "fontSize (p. 261)": integer,
          "shadowXOffset (p. 261)": integer,
          "alignment (p. 261)": integer,
          "shadowColor (p. 261)": enum,
          "fontName (p. 261)": enum
        },
        "sccDestinationSettings (p. 247)": {
          "framerate (p. 340)": enum
        },
        "languageCode (p. 246)": enum,
        "customLanguageCode (p. 246)": "string"
      }
    }
  }
}
"nameModifier (p. 333)" : "string",
],
"outputGroupSettings (p. 334)" : {
  "dashIsoGroupSettings (p. 334)" : {
    "fragmentLength (p. 255)" : integer,
    "baseUrl (p. 255)" : "string",
    "minBufferTime (p. 255)" : integer,
    "encryption (p. 255)" : {
      "spekeKeyProvider (p. 255)" : {
        "resourceId (p. 340)" : "string",
        "systemIds (p. 341)" : [
          "string"
        ],
        "url (p. 341)" : "string"
      }
    },
    "destination (p. 256)" : "string",
    "segmentLength (p. 256)" : integer,
    "segmentControl (p. 256)" : enum,
    "hbbtvCompliance (p. 256)" : enum
  },
  "fileGroupSettings (p. 334)" : {
    "destination (p. 269)" : "string"
  },
  "msSmoothGroupSettings (p. 335)" : {
    "fragmentLength (p. 330)" : integer,
    "encryption (p. 330)" : {
      "spekeKeyProvider (p. 330)" : {
        "resourceId (p. 340)" : "string",
        "systemIds (p. 341)" : [
          "string"
        ],
        "url (p. 341)" : "string"
      }
    },
    "audioDeduplication (p. 330)" : enum,
    "manifestEncoding (p. 330)" : enum,
    "destination (p. 330)" : "string"
  },
  "cmafGroupSettings (p. 335)" : {
    "writeHlsManifest (p. 249)" : enum,
    "writeDashManifest (p. 249)" : enum,
    "fragmentLength (p. 249)" : integer,
    "streamInfResolution (p. 250)" : enum,
    "minBufferTime (p. 250)" : integer,
    "clientCache (p. 250)" : enum,
    "codecSpecification (p. 250)" : enum,
    "destination (p. 250)" : "string",
    "segmentControl (p. 250)" : enum,
    "baseUrl (p. 250)" : "string",
    "encryption (p. 250)" : {
      "initializationVectorInManifest (p. 248)" : enum,
      "constantInitializationVector (p. 249)" : "string",
      "staticKeyProvider (p. 249)" : {
        "keyFormatVersions (p. 341)" : "string",
        "keyFormat (p. 341)" : "string",
        "staticKeyValue (p. 341)" : "string",
        "url (p. 341)" : "string"
      },
      "type (p. 249)" : enum,
      "encryptionMethod (p. 249)" : enum
    },
    "manifestCompression (p. 251)" : enum,
    "segmentLength (p. 251)" : integer,
    "manifestDurationFormat (p. 251)" : enum
"type": enum,
"hlsGroupSettings": {
  "segmentsPerSubdirectory": integer,
  "streamInfResolution": enum,
  "timestampDeltaMilliseconds": integer,
  "outputSelection": enum,
  "captionLanguageMappings": [
    {
      "languageDescription": "string",
      "captionChannel": integer,
      "languageCode": enum,
      "customLanguageCode": "string"
    }
  ],
  "clientCache": enum,
  "codecSpecification": enum,
  "destination": "string",
  "timedMetadataId3Frame": enum,
  "segmentControl": enum,
  "timedMetadataId3Period": integer,
  "captionLanguageSetting": enum,
  "minSegmentLength": integer,
  "directoryStructure": enum,
  "programDateTime": enum,
  "baseUrl": "string",
  "encryption": {
    "initializationVectorInManifest": enum,
    "constantInitializationVector": enum,
    "staticKeyProvider": {
      "keyFormatVersions": "string",
      "keyFormat": "string",
      "staticKeyValue": "string",
      "url": "string"
    },
    "type": enum,
    "encryptionMethod": enum,
    "spekeKeyProvider": {
      "resourceId": "string",
      "systemIds": ["string"
      ],
      "url": "string"
    }
  },
  "adMarkers": enum,
  "programDateTimePeriod": integer,
  "manifestDurationFormat": enum
},
"name": "string",
"customName": "string"
},
"nielsenConfiguration": {
  "distributorId": "string",
  "breakoutCode": integer
},
"inputs": [
  {
    "audioSelectors": []
  }
]
"audioSelectorGroups (p. 301)": {
},
"filterEnable (p. 301)": enum,
"deblockFilter (p. 301)": enum,
"videoSelector (p. 301)": {
  "colorSpace (p. 349)": enum,
  "hdr10Metadata (p. 349)": {
    "redPrimaryY (p. 289)": integer,
    "greenPrimaryY (p. 289)": integer,
    "whitePointX (p. 289)": integer,
    "maxLuminance (p. 289)": integer,
    "greenPrimaryX (p. 290)": integer,
    "whitePointY (p. 290)": integer,
    "redPrimaryX (p. 290)": integer,
    "bluePrimaryX (p. 290)": integer,
    "bluePrimaryY (p. 290)": integer,
    "maxFrameAverageLightLevel (p. 291)": integer,
    "maxContentLightLevel (p. 291)": integer,
    "minLuminance (p. 291)": integer
  },
  "programNumber (p. 349)": integer,
  "pid (p. 349)": integer,
  "colorSpaceUsage (p. 350)": enum
},
"filterStrength (p. 301)": integer,
"programNumber (p. 301)": integer,
"timecodeSource (p. 301)": enum,
"denoiseFilter (p. 301)": enum,
"captionSelectors (p. 302)": {
},
"inputClippings (p. 302)": [
  {
    "startTimecode (p. 299)": "string",
    "endTimecode (p. 299)": "string"
  }
],
"psiControl (p. 302)": enum
},
"timedMetadataInsertion (p. 306)": {
  "id3Insertions (p. 344)": [
    {
      "id3 (p. 299)": "string",
      "timecode (p. 299)": "string"
    }
  ]
},
"availBlanking (p. 306)": {
  "availBlankingImage (p. 241)": "string"
},
"name (p. 305)": "string",
"description (p. 305)": "string",
"arn (p. 305)": "string",
"category (p. 305)": "string",
"type (p. 305)": enum,
"queue (p. 305)": "string"
}

Example DeleteJobTemplateResponse

{
}

231
Example ExceptionBody

```json
{
  "message (p. 269)": "string"
}
```

Properties

**AacAudioDescriptionBroadcasterMix (enum)**

Choose BROADCASTER_MIXED_AD when the input contains pre-mixed main audio + audio description (AD) as a stereo pair. The value for AudioType will be set to 3, which signals to downstream systems that this stream contains "broadcaster mixed AD". Note that the input received by the encoder must contain pre-mixed audio; the encoder does not perform the mixing. When you choose BROADCASTER_MIXED_AD, the encoder ignores any values you provide in AudioType and FollowInputAudioType. Choose NORMAL when the input does not contain pre-mixed audio + audio description (AD). In this case, the encoder will use any values you provide for AudioType and FollowInputAudioType.

- BROADCASTER_MIXED_AD
- NORMAL

**AacCodecProfile (enum)**

AAC Profile.

- LC
- HEV1
- HEV2

**AacCodingMode (enum)**

Mono (Audio Description), Mono, Stereo, or 5.1 channel layout. Valid values depend on rate control mode and profile. "1.0 - Audio Description (Receiver Mix)" setting receives a stereo description plus control track and emits a mono AAC encode of the description track, with control data emitted in the PES header as per ETSI TS 101 154 Annex E.

- AD_RECEIVER_MIX
- CODING_MODE_1_0
- CODING_MODE_1_1
- CODING_MODE_2_0
- CODING_MODE_5_1

**AacRateControlMode (enum)**

Rate Control Mode.

- CBR
- VBR

**AacRawFormat (enum)**

Enables LATM/LOAS AAC output. Note that if you use LATM/LOAS AAC in an output, you must choose "No container" for the output container.
AacSettings

vbrQuality

Type: string
Required: False

codecProfile

Type: string
Required: False

codingMode

Type: string
Required: True

specification

Type: string
Required: False

bitrate

Average bitrate in bits/second. Defaults and valid values depend on rate control mode and profile.

Type: integer
Required: False
Minimum: 6000
Maximum: 1024000

rawFormat

Type: string
Required: False

rateControlMode

Type: string
Required: False

sampleRate

Sample rate in Hz. Valid values depend on rate control mode and profile.

Type: integer
Required: True
Minimum: 8000
Maximum: 96000

**audioDescriptionBroadcasterMix**

**Type**: string  
**Required**: False

**AacSpecification (enum)**

Use MPEG-2 AAC instead of MPEG-4 AAC audio for raw or MPEG-2 Transport Stream containers.

- MPEG2
- MPEG4

**AacVbrQuality (enum)**

VBR Quality Level - Only used if rate_control_mode is VBR.

- LOW
- MEDIUM_LOW
- MEDIUM_HIGH
- HIGH

**Ac3BitstreamMode (enum)**

Specifies the "Bitstream Mode" (bsmod) for the emitted AC-3 stream. See ATSC A/52-2012 for background on these values.

- COMPLETE_MAIN
- COMMENTARY
- DIALOGUE
- EMERGENCY
- HEARING_IMPAIRED
- MUSIC_AND_EFFECTS
- VISUALLY_IMPAIRED
- VOICE_OVER

**Ac3CodingMode (enum)**

Dolby Digital coding mode. Determines number of channels.

- CODING_MODE_1_0
- CODING_MODE_1_1
- CODING_MODE_2_0
- CODING_MODE_3_2_LFE

**Ac3DynamicRangeCompressionProfile (enum)**

If set to FILM_STANDARD, adds dynamic range compression signaling to the output bitstream as defined in the Dolby Digital specification.

- FILM_STANDARD
- NONE
Ac3LfeFilter (enum)

Applies a 120Hz lowpass filter to the LFE channel prior to encoding. Only valid with 3_2_LFE coding mode.

- ENABLED
- DISABLED

Ac3MetadataControl (enum)

When set to FOLLOW_INPUT, encoder metadata will be sourced from the DD, DD+, or DolbyE decoder that supplied this audio data. If audio was not supplied from one of these streams, then the static metadata settings will be used.

- FOLLOW_INPUT
- USE_CONFIGURED

Ac3Settings

dynamicRangeCompressionProfile

Type: string
Required: False

dialnorm

Sets the dialnorm for the output. If blank and input audio is Dolby Digital, dialnorm will be passed through.

Type: integer
Required: False
Minimum: 1
Maximum: 31

codingMode

Type: string
Required: False

metadataControl

Type: string
Required: False

bitrate

Average bitrate in bits/second. Valid bitrates depend on the coding mode.

Type: integer
Required: False
Minimum: 64000
Maximum: 640000
Properties

lfeFilter
- **Type**: string
- **Required**: False

bitstreamMode
- **Type**: string
- **Required**: False

sampleRate
Sample rate in hz. Sample rate is always 48000.
- **Type**: integer
- **Required**: False
- **Minimum**: 48000
- **Maximum**: 48000

AfdSignaling (enum)
This setting only applies to H.264 and MPEG2 outputs. Use Insert AFD signaling (AfdSignaling) to specify whether the service includes AFD values in the output video data and what those values are. * Choose None to remove all AFD values from this output. * Choose Fixed to ignore input AFD values and instead encode the value specified in the job. * Choose Auto to calculate output AFD values based on the input AFD scaler data.
- **NONE**
- **AUTO**
- **FIXED**

AiffSettings

cannels
Set Channels to specify the number of channels in this output audio track. Choosing Mono in the console will give you 1 output channel; choosing Stereo will give you 2. In the API, valid values are 1 and 2.
- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 2

bitDepth
Specify Bit depth (BitDepth), in bits per sample, to choose the encoding quality for this audio track.
- **Type**: integer
- **Required**: False
- **Minimum**: 16
- **Maximum**: 24

sampleRate
Sample rate in hz.
Properties

**Type**: integer  
**Required**: False  
**Minimum**: 8000  
**Maximum**: 192000

**AncillarySourceSettings**

**sourceAncillaryChannelNumber**

Specifies the 608 channel number in the ancillary data track from which to extract captions. Unused for passthrough.

**Type**: integer  
**Required**: False  
**Minimum**: 1  
**Maximum**: 4

**AntiAlias (enum)**

Enable Anti-alias (AntiAlias) to enhance sharp edges in video output when your input resolution is much larger than your output resolution. Default is enabled.

- DISABLED  
- ENABLED

**AudioCodec (enum)**

Type of Audio codec.

- AAC  
- MP2  
- WAV  
- AIFF  
- AC3  
- EAC3  
- PASSTHROUGH

**AudioCodecSettings**

**codec**

**Type**: string  
**Required**: True

**wavSettings**

**Type**: WavSettings (p. 350)  
**Required**: False

**aacSettings**

**Type**: AacSettings (p. 233)  
**Required**: False
ac3Settings
   Type: Ac3Settings (p. 235)
   Required: False

aiffSettings
   Type: AiffSettings (p. 236)
   Required: False

eac3Settings
   Type: Eac3Settings (p. 265)
   Required: False

mp2Settings
   Type: Mp2Settings (p. 321)
   Required: False

AudioDefaultSelection (enum)
Enable this setting on one audio selector to set it as the default for the job. The service uses this default for outputs where it can't find the specified input audio. If you don't set a default, those outputs have no audio.

   DEFAULT
   NOT_DEFAULT

AudioDescription

audioTypeControl
   Type: string
   Required: False

languageCodeControl
   Type: string
   Required: False

remixSettings
Advanced audio remixing settings.

   Type: RemixSettings (p. 339)
   Required: False

audioType
Applies only if Follow Input Audio Type is unchecked (false). A number between 0 and 255. The following are defined in ISO-IEC 13818-1: 0 = Undefined, 1 = Clean Effects, 2 = Hearing Impaired, 3 = Visually Impaired Commentary, 4-255 = Reserved.
Type: integer
Required: False
Minimum: 0
Maximum: 255

**audioSourceName**

Specifies which audio data to use from each input. In the simplest case, specify an "Audio Selector":#inputs-audio_selector by name based on its order within each input. For example if you specify "Audio Selector 3", then the third audio selector will be used from each input. If an input does not have an "Audio Selector 3", then the audio selector marked as "default" in that input will be used. If there is no audio selector marked as "default", silence will be inserted for the duration of that input. Alternatively, an "Audio Selector Group":#inputs-audio_selector_group name may be specified, with similar default/silence behavior. If no audio_source_name is specified, then "Audio Selector 1" will be chosen automatically.

Type: string
Required: False

**codecSettings**

Type: AudioCodecSettings (p. 237)
Required: True

**languageCode**

Indicates the language of the audio output track. The ISO 639 language specified in the 'Language Code' drop down will be used when 'Follow Input Language Code' is not selected or when 'Follow Input Language Code' is selected but there is no ISO 639 language code specified by the input.

Type: string
Required: False

**streamName**

Used for MS Smooth and Apple HLS outputs. Indicates the name displayed by the player (eg. English, or Director Commentary). Alphanumeric characters, spaces, and underscore are legal.

Type: string
Required: False
Pattern: ^[\w\s]*$

**audioNormalizationSettings**

Type: AudioNormalizationSettings (p. 240)
Required: False

**customLanguageCode**

Specify the language for this audio output track, using the ISO 639-2 or ISO 639-3 three-letter language code. The language specified will be used when 'Follow Input Language Code' is not selected or when 'Follow Input Language Code' is selected but there is no ISO 639 language code specified by the input.

Type: string
Required: False
**Pattern:** ^[A-Za-z]{3}$

**AudioLanguageCodeControl (enum)**
Choosing FOLLOW_INPUT will cause the ISO 639 language code of the output to follow the ISO 639 language code of the input. The language specified for languageCode' will be used when USE_CONFIGURED is selected or when FOLLOW_INPUT is selected but there is no ISO 639 language code specified by the input.

- FOLLOW_INPUT
- USE_CONFIGURED

**AudioNormalizationAlgorithm (enum)**
Audio normalization algorithm to use. 1770-1 conforms to the CALM Act specification, 1770-2 conforms to the EBU R-128 specification.

- ITU_BS_1770_1
- ITU_BS_1770_2

**AudioNormalizationAlgorithmControl (enum)**
When enabled the output audio is corrected using the chosen algorithm. If disabled, the audio will be measured but not adjusted.

- CORRECT_AUDIO
- MEASURE_ONLY

**AudioNormalizationLoudnessLogging (enum)**
If set to LOG, log each output's audio track loudness to a CSV file.

- LOG
- DONT_LOG

**AudioNormalizationPeakCalculation (enum)**
If set to TRUE_PEAK, calculate and log the TruePeak for each output's audio track loudness.

- TRUE_PEAK
- NONE

**AudioNormalizationSettings**

**targetLkfs**
Target LKFS(loudness) to adjust volume to. If no value is entered, a default value will be used according to the chosen algorithm. The CALM Act (1770-1) recommends a target of -24 LKFS. The EBU R-128 specification (1770-2) recommends a target of -23 LKFS.

- **Type:** number
- **Required:** False
- **Format:** float
- **Minimum:** -59.0
- **Maximum:** 0.0
algorithmControl

- **Type**: string
- **Required**: False

loudnessLogging

- **Type**: string
- **Required**: False

peakCalculation

- **Type**: string
- **Required**: False

correctionGateLevel

Content measuring above this level will be corrected to the target level. Content measuring below this level will not be corrected. Gating only applies when not using real_time_correction.

- **Type**: integer
- **Required**: False
- **Minimum**: -70
- **Maximum**: 0

algorithm

- **Type**: string
- **Required**: False

AudioSelectorType (enum)

Specifies the type of the audio selector.

- PID
- TRACK
- LANGUAGE_CODE

AudioTypeControl (enum)

When set to FOLLOW_INPUT, if the input contains an ISO 639 audio_type, then that value is passed through to the output. If the input contains no ISO 639 audio_type, the value in Audio Type is included in the output. Otherwise the value in Audio Type is included in the output. Note that this field and audioType are both ignored if audioDescriptionBroadcasterMix is set to BROADCASTER_MIXED_AD.

- FOLLOW_INPUT
- USE_CONFIGURED

AvailBlanking

availBlankingImage

Blanking image to be used. Leave empty for solid black. Only bmp and png images are supported.
Properties

Type: string
Required: False
Pattern: ^(s3://\/
(.*?)\.(bmp|BMP|png|PNG)$

BurninDestinationSettings

xPosition

Specifies the horizontal position of the caption relative to the left side of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the left of the output. If no explicit x_position is provided, the horizontal caption position will be determined by the alignment parameter. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

Type: integer
Required: False
Minimum: 0
Maximum: 2147483647

BackgroundColor

Type: string
Required: False

TeletextSpacing

Type: string
Required: False

yPosition

Specifies the vertical position of the caption relative to the top of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the top of the output. If no explicit y_position is provided, the caption will be positioned towards the bottom of the output. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

Type: integer
Required: False
Minimum: 0
Maximum: 2147483647

BackgroundOpacity

Specifies the opacity of the background rectangle. 255 is opaque; 0 is transparent. Leaving this parameter blank is equivalent to setting it to 0 (transparent). All burn-in and DVB-Sub font settings must match.

Type: integer
Required: False
Minimum: 0
Maximum: 255
fontOpacity
Specifies the opacity of the burned-in captions. 255 is opaque; 0 is transparent. All burn-in and DVB-Sub font settings must match.

Type: integer
Required: True
Minimum: 0
Maximum: 255

fontResolution
Font resolution in DPI (dots per inch); default is 96 dpi. All burn-in and DVB-Sub font settings must match.

Type: integer
Required: False
Minimum: 96
Maximum: 600

shadowOpacity
Specifies the opacity of the shadow. 255 is opaque; 0 is transparent. Leaving this parameter blank is equivalent to setting it to 0 (transparent). All burn-in and DVB-Sub font settings must match.

Type: integer
Required: False
Minimum: 0
Maximum: 255

shadowYOffset
Specifies the vertical offset of the shadow relative to the captions in pixels. A value of -2 would result in a shadow offset 2 pixels above the text. All burn-in and DVB-Sub font settings must match.

Type: integer
Required: False
Minimum: -2147483648
Maximum: 2147483647

outlineSize
Specifies font outline size in pixels. This option is not valid for source captions that are either 608/ embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

Type: integer
Required: True
Minimum: 0
Maximum: 10

outlineColor
Type: string
Required: True
**fontSize**

A positive integer indicates the exact font size in points. Set to 0 for automatic font size selection. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 96

**shadowXOffset**

Specifies the horizontal offset of the shadow relative to the captions in pixels. A value of -2 would result in a shadow offset 2 pixels to the left. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: False
- **Minimum**: -2147483648
- **Maximum**: 2147483647

**alignment**

- **Type**: string
- **Required**: True

**shadowColor**

- **Type**: string
- **Required**: False

**fontColor**

- **Type**: string
- **Required**: False

**BurninSubtitleAlignment (enum)**

If no explicit x_position or y_position is provided, setting alignment to centered will place the captions at the bottom center of the output. Similarly, setting a left alignment will align captions to the bottom left of the output. If x and y positions are given in conjunction with the alignment parameter, the font will be justified (either left or centered) relative to those coordinates. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- CENTERED
- LEFT

**BurninSubtitleBackgroundColor (enum)**

Specifies the color of the rectangle behind the captions. All burn-in and DVB-Sub font settings must match.

- NONE
- BLACK
**WHITE**

**BurninSubtitleFontColor (enum)**

Specifies the color of the burned-in captions. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- WHITE
- BLACK
- YELLOW
- RED
- GREEN
- BLUE

**BurninSubtitleOutlineColor (enum)**

Specifies font outline color. This option is not valid for source captions that are either 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- BLACK
- WHITE
- YELLOW
- RED
- GREEN
- BLUE

**BurninSubtitleShadowColor (enum)**

Specifies the color of the shadow cast by the captions. All burn-in and DVB-Sub font settings must match.

- NONE
- BLACK
- WHITE

**BurninSubtitleTeletextSpacing (enum)**

Only applies to jobs with input captions in Teletext or STL formats. Specify whether the spacing between letters in your captions is set by the captions grid or varies depending on letter width. Choose fixed grid to conform to the spacing specified in the captions file more accurately. Choose proportional to make the text easier to read if the captions are closed caption.

- FIXED_GRID
- PROPORTIONAL

**CaptionDescription**

**captionSelectorName**

Specifies which "Caption Selector":#inputs-caption_selector to use from each input when generating captions. The name should be of the format "Caption Selector <N>", which denotes that the Nth Caption Selector will be used from each input.
Type: string
Required: True

**languageDescription**

Human readable information to indicate captions available for players (e.g. English, or Spanish). Alphanumeric characters, spaces, and underscore are legal.

Type: string
Required: False
Pattern: ^[\w \]*$

**destinationSettings**

Type: CaptionDestinationSettings (p. 246)
Required: True

**languageCode**

Indicates the language of the caption output track.

Type: string
Required: False

**customLanguageCode**

Indicates the language of the caption output track, using the ISO 639-2 or ISO 639-3 three-letter language code.

Type: string
Required: False
Pattern: ^[A-Za-z]{3}$

**CaptionDestinationSettings**

**burninDestinationSettings**

Type: BurninDestinationSettings (p. 242)
Required: False

**teletextDestinationSettings**

Type: TeletextDestinationSettings (p. 342)
Required: False

**ttmlDestinationSettings**

Type: TtmlDestinationSettings (p. 344)
Required: False

**destinationType**

Type: string
Required: True

dvbSubDestinationSettings

Type: DvbSubDestinationSettings (p. 259)
Required: False

sccDestinationSettings

Type: SccDestinationSettings (p. 340)
Required: False

CaptionDestinationType (enum)

Type of Caption output, including Burn-In, Embedded, SCC, SRT, TTML, WebVTT, DVB-Sub, Teletext.

BURN_IN
DVB_SUB
EMBEDDED
SCC
SRT
TELETEXT
TTML
WEBVTT

CaptionSourceSettings

fileSourceSettings

Type: FileSourceSettings (p. 270)
Required: False

ancillarySourceSettings

Type: AncillarySourceSettings (p. 237)
Required: False

embeddedSourceSettings

Type: EmbeddedSourceSettings (p. 268)
Required: False

sourceType

Type: string
Required: True

dvbSubSourceSettings

Type: DvbSubSourceSettings (p. 262)
Required: False

**teletextSourceSettings**

*Type:* `TeletextSourceSettings (p. 342)`
*Required:* False

**CaptionSourceType (enum)**

Use `SourceType` to identify the format of your input captions. The service cannot auto-detect caption format.

- ANCILLARY
- DVB_SUB
- EMBEDDED
- SCC
- TTML
- STL
- SRT
- TELETEXT
- NULL_SOURCE

**ChannelMapping**

**outputChannels**

List of output channels

*Type:* Array of type `OutputChannelMapping (p. 334)`
*Required:* True

**CmafClientCache (enum)**

When set to ENABLED, sets #EXT-X-ALLOW-CACHE:no tag, which prevents client from saving media segments for later replay.

- DISABLED
- ENABLED

**CmafCodecSpecification (enum)**

Specification to use (RFC-6381 or the default RFC-4281) during m3u8 playlist generation.

- RFC_6381
- RFC_4281

**CmafEncryptionSettings**

**initializationVectorInManifest**

*Type:* string
*Required:* False
constantInitializationVector
This is a 128-bit, 16-byte hex value represented by a 32-character text string. If this parameter is not set then the Initialization Vector will follow the segment number by default.

  Type: string
  Required: False
  Pattern: ^[0-9a-fA-F]{32}$

staticKeyProvider

  Type: StaticKeyProvider (p. 341)
  Required: False

type

  Type: string
  Required: True

encryptionMethod

  Type: string
  Required: False

CmafEncryptionType (enum)
Encrypts the segments with the given encryption scheme. Leave blank to disable. Selecting 'Disabled' in the web interface also disables encryption.

  SAMPLE_AES

CmafGroupSettings

writeHlsManifest

  Type: string
  Required: False

writeDashManifest

  Type: string
  Required: False

fragmentLength
Length of fragments to generate (in seconds). Fragment length must be compatible with GOP size and Framerate. Note that fragments will end on the next keyframe after this number of seconds, so actual fragment length may be longer. When Emit Single File is checked, the fragmentation is internal to a single output file and it does not cause the creation of many output files as in other output types.

  Type: integer
  Required: True
  Minimum: 1
  Maximum: 2147483647
streamInfResolution

Type: string
Required: False

minBufferTime

Minimum time of initially buffered media that is needed to ensure smooth playout.

Type: integer
Required: False
Minimum: 0
Maximum: 2147483647

clientCache

Type: string
Required: False

codecSpecification

Type: string
Required: False

destination

Use Destination (Destination) to specify the S3 output location and the output filename base. Destination accepts format identifiers. If you do not specify the base filename in the URI, the service will use the filename of the input file. If your job has multiple inputs, the service uses the filename of the first input file.

Type: string
Required: False
Pattern: ^s3:\/\/

segmentControl

Type: string
Required: False

baseUrl

A partial URI prefix that will be put in the manifest file at the top level BaseURL element. Can be used if streams are delivered from a different URL than the manifest file.

Type: string
Required: False

encryption

DRM settings.

Type: CmafEncryptionSettings (p. 248)
**Properties**

**manifestCompression**

*Type:* string  
*Required:* False

**segmentLength**

Use this setting to specify the length, in seconds, of each individual CMAF segment. This value applies to the whole package; that is, to every output in the output group. Note that segments end on the first keyframe after this number of seconds, so the actual segment length might be slightly longer. If you set Segment control (CmafSegmentControl) to single file, the service puts the content of each output in a single file that has metadata that marks these segments. If you set it to segmented files, the service creates multiple files for each output, each with the content of one segment.

*Type:* integer  
*Required:* True  
*Minimum:* 1  
*Maximum:* 2147483647

**manifestDurationFormat**

*Type:* string  
*Required:* False

**CmafInitializationVectorInManifest (enum)**

The Initialization Vector is a 128-bit number used in conjunction with the key for encrypting blocks. If set to INCLUDE, Initialization Vector is listed in the manifest. Otherwise Initialization Vector is not in the manifest.

INCLUDE  
EXCLUDE

**CmafKeyProviderType (enum)**

Indicates which type of key provider is used for encryption.

STATIC_KEY

**CmafManifestCompression (enum)**

When set to GZIP, compresses HLS playlist.

GZIP  
NONE

**CmafManifestDurationFormat (enum)**

Indicates whether the output manifest should use floating point values for segment duration.

FLOATING_POINT  
INTEGER
**CmafSegmentControl (enum)**

When set to SINGLE_FILE, a single output file is generated, which is internally segmented using the Fragment Length and Segment Length. When set to SEGMENTED_FILES, separate segment files will be created.

- SINGLE_FILE
- SEGMENTED_FILES

**CmafStreamInfResolution (enum)**

Include or exclude RESOLUTION attribute for video in EXT-X-STREAM-INF tag of variant manifest.

- INCLUDE
- EXCLUDE

**CmafWriteDASHManifest (enum)**

When set to ENABLED, a DASH MPD manifest will be generated for this output.

- DISABLED
- ENABLED

**CmafWriteHLSManifest (enum)**

When set to ENABLED, an Apple HLS manifest will be generated for this output.

- DISABLED
- ENABLED

**ColorCorrector**

**saturation**

Saturation level.

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 100

**brightness**

Brightness level.

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 100

**hdr10Metadata**

- **Type**: Hdr10Metadata (p. 289)
- **Required**: False
contrast

Contrast level.

Type: integer  
Required: False  
Minimum: 1  
Maximum: 100

hue

Hue in degrees.

Type: integer  
Required: False  
Minimum: -180  
Maximum: 180

colorSpaceConversion

Type: string  
Required: False

ColorMetadata (enum)

Enable insert color metadata (ColorMetadata) to include color metadata in this output. This setting is enabled by default.

IGNORE
INSERT

ColorSpace (enum)

If your input video has accurate color space metadata, or if you don't know about color space, leave this set to the default value FOLLOW. The service will automatically detect your input color space. If your input video has metadata indicating the wrong color space, or if your input video is missing color space metadata that should be there, specify the accurate color space here. If you choose HDR10, you can also correct inaccurate color space coefficients, using the HDR master display information controls. You must also set Color space usage (ColorSpaceUsage) to FORCE for the service to use these values.

FOLLOW
REC_601
REC_709
HDR10
HLG_2020

ColorSpaceConversion (enum)

Determines if colorspace conversion will be performed. If set to _None_, no conversion will be performed. If _Force 601_ or _Force 709_ are selected, conversion will be performed for inputs with differing colorspace. An input's colorspace can be specified explicitly in the "Video Selector":#inputs-video_selector if necessary.

NONE
FORCE_601
FORCE_709
FORCE_HDR10
FORCE_HLG_2020

**ColorSpaceUsage (enum)**

There are two sources for color metadata, the input file and the job configuration (in the Color space and HDR master display information settings). The Color space usage setting controls which takes precedence. FORCE: The system will use color metadata supplied by user, if any. If the user does not supply color metadata, the system will use data from the source. FALLBACK: The system will use color metadata from the source. If source has no color metadata, the system will use user-supplied color metadata values if available.

FORCE
FALLBACK

**ContainerSettings**

**container**

*Type:* string  
*Required:* True

**mp4Settings**

*Type:* Mp4Settings (p. 322)  
*Required:* False

**m3u8Settings**

*Type:* M3u8Settings (p. 317)  
*Required:* False

**m2tsSettings**

*Type:* M2tsSettings (p. 312)  
*Required:* False

**movSettings**

*Type:* MovSettings (p. 321)  
*Required:* False

**f4vSettings**

*Type:* F4vSettings (p. 269)  
*Required:* False

**ContainerType (enum)**

Container for this output. Some containers require a container settings object. If not specified, the default object will be created.
DashIsoEncryptionSettings

spekeKeyProvider

Type: SpekeKeyProvider (p. 340)
Required: True

DashIsoGroupSettings

fragmentLength

Length of fragments to generate (in seconds). Fragment length must be compatible with GOP size and Framerate. Note that fragments will end on the next keyframe after this number of seconds, so actual fragment length may be longer. When Emit Single File is checked, the fragmentation is internal to a single output file and it does not cause the creation of many output files as in other output types.

Type: integer
Required: True
Minimum: 1
Maximum: 2147483647

baseUrl

A partial URI prefix that will be put in the manifest (.mpd) file at the top level BaseURL element. Can be used if streams are delivered from a different URL than the manifest file.

Type: string
Required: False

minBufferTime

Minimum time of initially buffered media that is needed to ensure smooth playout.

Type: integer
Required: False
Minimum: 0
Maximum: 2147483647

encryption

DRM settings.

Type: DashIsoEncryptionSettings (p. 255)
Required: False
**destination**

Use Destination (Destination) to specify the S3 output location and the output filename base. Destination accepts format identifiers. If you do not specify the base filename in the URI, the service will use the filename of the input file. If your job has multiple inputs, the service uses the filename of the first input file.

*Type: string*  
*Required: False*  
*Pattern: ^s3:/

**segmentLength**

Length of mpd segments to create (in seconds). Note that segments will end on the next keyframe after this number of seconds, so actual segment length may be longer. When Emit Single File is checked, the segmentation is internal to a single output file and it does not cause the creation of many output files as in other output types.

*Type: integer*  
*Required: True*  
*Minimum: 1*  
*Maximum: 2147483647*

**segmentControl**

*Type: string*  
*Required: False*

**hbbtvCompliance**

*Type: string*  
*Required: False*

**DashIsoHbbtvCompliance (enum)**

Supports HbbTV specification as indicated

- HBTTV_1_5
- NONE

**DashIsoSegmentControl (enum)**

When set to SINGLE_FILE, a single output file is generated, which is internally segmented using the Fragment Length and Segment Length. When set to SEGMENTED_FILES, separate segment files will be created.

- SINGLE_FILE
- SEGMENTED_FILES

**DeinterlaceAlgorithm (enum)**

Only applies when you set Deinterlacer (DeinterlaceMode) to Deinterlace (DEINTERLACE) or Adaptive (ADAPTIVE). Motion adaptive interpolate (INTERPOLATE) produces sharper pictures, while blend (BLEND)
produces smoother motion. Use (INTERPOLATE_TICKER) OR (BLEND_TICKER) if your source file includes a ticker, such as a scrolling headline at the bottom of the frame.

INTERPOLATE
INTERPOLATE_TICKER
BLEND
BLEND_TICKER

Deinterlacer

mode

Type: string
Required: False

control

Type: string
Required: False

algorithm

Type: string
Required: False

DeinterlacerControl (enum)

- When set to NORMAL (default), the deinterlacer does not convert frames that are tagged in metadata as progressive. It will only convert those that are tagged as some other type. - When set to FORCE_ALL_FRAMES, the deinterlacer converts every frame to progressive - even those that are already tagged as progressive. Turn Force mode on only if there is a good chance that the metadata has tagged frames as progressive when they are not progressive. Do not turn on otherwise; processing frames that are already progressive into progressive will probably result in lower quality video.

FORCE_ALL_FRAMES
NORMAL

DeinterlacerMode (enum)

Use Deinterlacer (DeinterlaceMode) to choose how the service will do deinterlacing. Default is Deinterlace. - Deinterlace converts interlaced to progressive. - Inverse telecine converts Hard Telecine 29.97i to progressive 23.976p. - Adaptive auto-detects and converts to progressive.

DEINTERLACE
INVERSE_TELECINE
ADAPTIVE

DeleteJobTemplateRequest

name

The name of the job template to be deleted.
DeleteJobTemplateResponse

DropFrameTimecode (enum)
Applies only to 29.97 fps outputs. When this feature is enabled, the service will use drop-frame timecode on outputs. If it is not possible to use drop-frame timecode, the system will fall back to non-drop-frame. This setting is enabled by default when Timecode insertion (TimecodeInsertion) is enabled.

- DISABLED
- ENABLED

DvbNitSettings

networkName
The network name text placed in the network_name_descriptor inside the Network Information Table. Maximum length is 256 characters.

- Type: string
- Required: True

networkId
The numeric value placed in the Network Information Table (NIT).

- Type: integer
- Required: True
- Minimum: 0
- Maximum: 65535

nitInterval
The number of milliseconds between instances of this table in the output transport stream.

- Type: integer
- Required: True
- Minimum: 25
- Maximum: 10000

DvbSdtSettings

sdtInterval
The number of milliseconds between instances of this table in the output transport stream.

- Type: integer
- Required: False
- Minimum: 25
- Maximum: 2000
**serviceName**

The service name placed in the service_descriptor in the Service Description Table. Maximum length is 256 characters.

- **Type**: string
- **Required**: False

**serviceProviderName**

The service provider name placed in the service_descriptor in the Service Description Table. Maximum length is 256 characters.

- **Type**: string
- **Required**: False

**outputSdt**

- **Type**: string
- **Required**: False

**DvbSubDestinationSettings**

**xPosition**

Specifies the horizontal position of the caption relative to the left side of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the left of the output. If no explicit x_position is provided, the horizontal caption position will be determined by the alignment parameter. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 2147483647

**backgroundColor**

- **Type**: string
- **Required**: False

**teletextSpacing**

- **Type**: string
- **Required**: False

**yPosition**

Specifies the vertical position of the caption relative to the top of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the top of the output. If no explicit y_position is provided, the caption will be positioned towards the bottom of the output. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.
Type: integer  
Required: False  
Minimum: 0  
Maximum: 2147483647

**backgroundOpacity**

Specifies the opacity of the background rectangle. 255 is opaque; 0 is transparent. Leaving this parameter blank is equivalent to setting it to 0 (transparent). All burn-in and DVB-Sub font settings must match.

Type: integer  
Required: False  
Minimum: 0  
Maximum: 255

**fontOpacity**

Specifies the opacity of the burned-in captions. 255 is opaque; 0 is transparent. All burn-in and DVB-Sub font settings must match.

Type: integer  
Required: True  
Minimum: 0  
Maximum: 255

**fontResolution**

Font resolution in DPI (dots per inch); default is 96 dpi. All burn-in and DVB-Sub font settings must match.

Type: integer  
Required: False  
Minimum: 96  
Maximum: 600

**shadowOpacity**

Specifies the opacity of the shadow. 255 is opaque; 0 is transparent. Leaving this parameter blank is equivalent to setting it to 0 (transparent). All burn-in and DVB-Sub font settings must match.

Type: integer  
Required: False  
Minimum: 0  
Maximum: 255

**shadowYOffset**

Specifies the vertical offset of the shadow relative to the captions in pixels. A value of -2 would result in a shadow offset 2 pixels above the text. All burn-in and DVB-Sub font settings must match.

Type: integer  
Required: False
**Properties**

**outlineSize**

Specifies font outline size in pixels. This option is not valid for source captions that are either 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

*Type:* integer  
*Required:* True  
*Minimum:* 0  
*Maximum:* 10

**outlineColor**

*Type:* string  
*Required:* True

**fontSize**

A positive integer indicates the exact font size in points. Set to 0 for automatic font size selection. All burn-in and DVB-Sub font settings must match.

*Type:* integer  
*Required:* False  
*Minimum:* 0  
*Maximum:* 96

**shadowXOffset**

Specifies the horizontal offset of the shadow relative to the captions in pixels. A value of -2 would result in a shadow offset 2 pixels to the left. All burn-in and DVB-Sub font settings must match.

*Type:* integer  
*Required:* False  
*Minimum:* -2147483648  
*Maximum:* 2147483647

**alignment**

*Type:* string  
*Required:* True

**shadowColor**

*Type:* string  
*Required:* False

**fontColor**

*Type:* string
Required: False

DvbSubSourceSettings

pid

When using DVB-Sub with Burn-In or SMPTE-TT, use this PID for the source content. Unused for DVB-Sub passthrough. All DVB-Sub content is passed through, regardless of selectors.

Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

DvbSubtitleAlignment (enum)

If no explicit x_position or y_position is provided, setting alignment to centered will place the captions at the bottom center of the output. Similarly, setting a left alignment will align captions to the bottom left of the output. If x and y positions are given in conjunction with the alignment parameter, the font will be justified (either left or centered) relative to those coordinates. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

CENTERED
LEFT

DvbSubtitleBackgroundColor (enum)

Specifies the color of the rectangle behind the captions. All burn-in and DVB-Sub font settings must match.

NONE
BLACK
WHITE

DvbSubtitleFontColor (enum)

Specifies the color of the burned-in captions. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

WHITE
BLACK
YELLOW
RED
GREEN
BLUE

DvbSubtitleOutlineColor (enum)

Specifies font outline color. This option is not valid for source captions that are either 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.
**DvbSubtitleShadowColor (enum)**

Specifies the color of the shadow cast by the captions. All burn-in and DVB-Sub font settings must match.

- NONE
- BLACK
- WHITE

**DvbSubtitleTeletextSpacing (enum)**

Only applies to jobs with input captions in Teletext or STL formats. Specify whether the spacing between letters in your captions is set by the captions grid or varies depending on letter width. Choose fixed grid to conform to the spacing specified in the captions file more accurately. Choose proportional to make the text easier to read if the captions are closed caption.

- FIXED_GRID
- PROPORTIONAL

**DvbTdtSettings**

**tdtInterval**

The number of milliseconds between instances of this table in the output transport stream.

- **Type:** integer
- **Required:** True
- **Minimum:** 1000
- **Maximum:** 30000

**Eac3AttenuationControl (enum)**

If set to ATTENUATE_3_DB, applies a 3 dB attenuation to the surround channels. Only used for 3/2 coding mode.

- ATTENUATE_3_DB
- NONE

**Eac3BitstreamMode (enum)**

Specifies the "Bitstream Mode" (bsmod) for the emitted E-AC-3 stream. See ATSC A/52-2012 (Annex E) for background on these values.

- COMPLETE_MAIN
- COMMENTARY
- EMERGENCY
HEARING_IMPAIRED
VISUALLY_IMPAIRED

Eac3CodingMode (enum)
Dolby Digital Plus coding mode. Determines number of channels.

CODING_MODE_1_0
CODING_MODE_2_0
CODING_MODE_3_2

Eac3DcFilter (enum)
Activates a DC highpass filter for all input channels.

ENABLED
DISABLED

Eac3DynamicRangeCompressionLine (enum)
Enables Dynamic Range Compression that restricts the absolute peak level for a signal.

NONE
FILM_STANDARD
FILM_LIGHT
MUSIC_STANDARD
MUSIC_LIGHT
SPEECH

Eac3DynamicRangeCompressionRf (enum)
Enables Heavy Dynamic Range Compression, ensures that the instantaneous signal peaks do not exceed specified levels.

NONE
FILM_STANDARD
FILM_LIGHT
MUSIC_STANDARD
MUSIC_LIGHT
SPEECH

Eac3LfeControl (enum)
When encoding 3/2 audio, controls whether the LFE channel is enabled

LFE
NO_LFE

Eac3LfeFilter (enum)
Applies a 120Hz lowpass filter to the LFE channel prior to encoding. Only valid with 3_2_LFE coding mode.

ENABLED
DISABLED

**Eac3MetadataControl (enum)**

When set to FOLLOW_INPUT, encoder metadata will be sourced from the DD, DD+, or DolbyE decoder that supplied this audio data. If audio was not supplied from one of these streams, then the static metadata settings will be used.

- FOLLOW_INPUT
- USE_CONFIGURED

**Eac3PassthroughControl (enum)**

When set to WHEN_POSSIBLE, input DD+ audio will be passed through if it is present on the input. This detection is dynamic over the life of the transcode. Inputs that alternate between DD+ and non-DD+ content will have a consistent DD+ output as the system alternates between passthrough and encoding.

- WHEN_POSSIBLE
- NO_PASSTHROUGH

**Eac3PhaseControl (enum)**

Controls the amount of phase-shift applied to the surround channels. Only used for 3/2 coding mode.

- SHIFT_90_DEGREES
- NO_SHIFT

**Eac3Settings**

**dialnorm**

Sets the dialnorm for the output. If blank and input audio is Dolby Digital Plus, dialnorm will be passed through.

- Type: integer
- Required: False
- Minimum: 1
- Maximum: 31

**passthroughControl**

- Type: string
- Required: False

**metadataControl**

- Type: string
- Required: False

**bitrate**

Average bitrate in bits/second. Valid bitrates depend on the coding mode.

- Type: integer
Properties

Required: False
Minimum: 64000
Maximum: 640000

dynamicRangeCompressionRf

Type: string
Required: False

sampleRate

Sample rate in hz. Sample rate is always 48000.

Type: integer
Required: False
Minimum: 48000
Maximum: 48000

surroundExMode

Type: string
Required: False

ltRtSurroundMixLevel

Left total/Right total surround mix level. Only used for 3/2 coding mode. Valid values: -1.5 -3.0 -4.5 -6.0 -60

Type: number
Required: False
Format: float
Minimum: -60.0
Maximum: -1.5

dynamicRangeCompressionLine

Type: string
Required: False

lfeControl

Type: string
Required: False

codingMode

Type: string
Required: False

surroundMode

Type: string
Required: False

attenuationControl
Type: string
Required: False

lfeFilter
Type: string
Required: False

phaseControl
Type: string
Required: False

ltRtCenterMixLevel
Left total/Right total center mix level. Only used for 3/2 coding mode. Valid values: 3.0, 1.5, 0.0, -1.5 -3.0 -4.5 -6.0 -60
Type: number
Required: False
Format: float
Minimum: -60.0
Maximum: 3.0

dcFilter
Type: string
Required: False

bitstreamMode
Type: string
Required: False

stereoDownmix
Type: string
Required: False

loRoSurroundMixLevel
Left only/Right only surround mix level. Only used for 3/2 coding mode. Valid values: -1.5 -3.0 -4.5 -6.0 -60
Type: number
Required: False
Format: float
Minimum: -60.0
Maximum: -1.5
**IoRoCenterMixLevel**

Left only/Right only center mix level. Only used for 3/2 coding mode. Valid values: 3.0, 1.5, 0.0, -1.5 -3.0 -4.5 -6.0 -60

- **Type**: number
- **Required**: False
- **Format**: float
- **Minimum**: -60.0
- **Maximum**: 3.0

**Eac3StereoDownmix (enum)**

Stereo downmix preference. Only used for 3/2 coding mode.

- **NOT_INDICATED**
- **LO_RO**
- **LT_RT**
- **DPL2**

**Eac3SurroundExMode (enum)**

When encoding 3/2 audio, sets whether an extra center back surround channel is matrix encoded into the left and right surround channels.

- **NOT_INDICATED**
- **ENABLED**
- **DISABLED**

**Eac3SurroundMode (enum)**

When encoding 2/0 audio, sets whether Dolby Surround is matrix encoded into the two channels.

- **NOT_INDICATED**
- **ENABLED**
- **DISABLED**

**EmbeddedConvert608To708 (enum)**

When set to UPCONVERT, 608 data is both passed through via the "608 compatibility bytes" fields of the 708 wrapper as well as translated into 708. 708 data present in the source content will be discarded.

- **UPCONVERT**
- **DISABLED**

**EmbeddedSourceSettings**

**source608ChannelNumber**

Specifies the 608/708 channel number within the video track from which to extract captions. Unused for passthrough.

- **Type**: integer
- **Required**: False
Minimum: 1
Maximum: 4

convert608To708
Type: string
Required: False

source608TrackNumber
Specifies the video track index used for extracting captions. The system only supports one input video track, so this should always be set to '1'.
Type: integer
Required: False
Minimum: 1
Maximum: 1

ExceptionBody

message
Type: string
Required: False

F4vMoovPlacement (enum)
If set to PROGRESSIVE_DOWNLOAD, the MOOV atom is relocated to the beginning of the archive as required for progressive downloading. Otherwise it is placed normally at the end.

PROGRESSIVE_DOWNLOAD
NORMAL

F4vSettings

moovPlacement
Type: string
Required: False

FileGroupSettings

destination
Use Destination (Destination) to specify the S3 output location and the output filename base. Destination accepts format identifiers. If you do not specify the base filename in the URI, the service will use the filename of the input file. If your job has multiple inputs, the service uses the filename of the first input file.
Type: string
Required: False
Pattern: ^s3:]//\//
**FileSourceConvert608To708 (enum)**

If set to UPCONVERT, 608 caption data is both passed through via the "608 compatibility bytes" fields of the 708 wrapper as well as translated into 708. 708 data present in the source content will be discarded.

- UPCONVERT
- DISABLED

**FileSourceSettings**

**timeDelta**

Specifies a time delta in seconds to offset the captions from the source file.

- **Type:** integer
- **Required:** False
- **Minimum:** -2147483648
- **Maximum:** 2147483647

**convert608To708**

- **Type:** string
- **Required:** False

**sourceFile**

External caption file used for loading captions. Accepted file extensions are 'scc', 'ttml', 'dfxp', 'stl', 'srt', and 'smi'.

- **Type:** string
- **Required:** True
- **Pattern:** ^s3:((.*)\.(scc|Scc|ttml|Ttml|dfxp|DFXP|stl|STL|srt|SRT|smi|SMI)$

**FrameCaptureSettings**

**framerateDenominator**

Frame capture will encode the first frame of the output stream, then one frame every framerateDenominator/framerateNumerator seconds. For example, settings of framerateNumerator = 1 and framerateDenominator = 3 (a rate of 1/3 frame per second) will capture the first frame, then 1 frame every 3s. Files will be named as filename.n.jpg where n is the 0-based sequence number of each Capture.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 2147483647

**maxCaptures**

Maximum number of captures (encoded jpg output files).

- **Type:** integer
- **Required:** False
- **Minimum:** 1
**Properties**

**framerateNumerator**
Frame capture will encode the first frame of the output stream, then one frame every framerateDenominator/framerateNumerator seconds. For example, settings of framerateNumerator = 1 and framerateDenominator = 3 (a rate of 1/3 frame per second) will capture the first frame, then 1 frame every 3s. Files will be named as filename.NNNNNNN.jpg where N is the 0-based frame sequence number zero padded to 7 decimal places.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 2147483647

**quality**
JPEG Quality - a higher value equals higher quality.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 100

**GetJobTemplateRequest**

**name**
The name of the job template.

- **Type:** string
- **Required:** False

**GetJobTemplateResponse**

**jobTemplate**

- **Type:** JobTemplate (p. 304)
- **Required:** False

**H264AdaptiveQuantization (enum)**
Adaptive quantization. Allows intra-frame quantizers to vary to improve visual quality.

- OFF
- LOW
- MEDIUM
- HIGH
- HIGHER
- MAX

**H264CodecLevel (enum)**
H.264 Level.
AUTO
LEVEL_1
LEVEL_1_1
LEVEL_1_2
LEVEL_1_3
LEVEL_2
LEVEL_2_1
LEVEL_2_2
LEVEL_3
LEVEL_3_1
LEVEL_3_2
LEVEL_4
LEVEL_4_1
LEVEL_4_2
LEVEL_5
LEVEL_5_1
LEVEL_5_2

H264CodecProfile (enum)

H.264 Profile. High 4:2:2 and 10-bit profiles are only available with the AVC-I License.

BASELINE
HIGH
HIGH_10BIT
HIGH_422
HIGH_422_10BIT
MAIN

H264EntropyEncoding (enum)

Entropy encoding mode. Use CABAC (must be in Main or High profile) or CAVLC.

CABAC
CAVLC

H264FieldEncoding (enum)

Choosing FORCE_FIELD disables PAFF encoding for interlaced outputs.

PAFF
FORCE_FIELD

H264FlickerAdaptiveQuantization (enum)

Adjust quantization within each frame to reduce flicker or 'pop' on I-frames.

DISABLED
ENABLED

H264FramerateControl (enum)

If you are using the console, use the Framerate setting to specify the framerate for this output. If you want to keep the same framerate as the input video, choose Follow source. If you want to do framerate conversion, choose a framerate from the dropdown list or choose Custom. The framerates

272
shown in the dropdown list are decimal approximations of fractions. If you choose Custom, specify your framerate as a fraction. If you are creating your transcoding job specification as a JSON file without the console, use FramerateControl to specify which value the service uses for the framerate for this output. Choose INITIALIZE_FROM_SOURCE if you want the service to use the framerate from the input. Choose SPECIFIED if you want the service to use the framerate you specify in the settings FramerateNumerator and FramerateDenominator.

INITIALIZE_FROM_SOURCE
SPECIFIED

H264FramerateConversionAlgorithm (enum)
When set to INTERPOLATE, produces smoother motion during framerate conversion.

DUPLICATE_DROP
INTERPOLATE

H264GopBReference (enum)
If enable, use reference B frames for GOP structures that have B frames > 1.

DISABLED
ENABLED

H264GopSizeUnits (enum)
Indicates if the GOP Size in H264 is specified in frames or seconds. If seconds the system will convert the GOP Size into a frame count at run time.

FRAMES
SECONDS

H264InterlaceMode (enum)
Use Interlace mode (InterlaceMode) to choose the scan line type for the output. * Top Field First (TOP_FIELD) and Bottom Field First (BOTTOM_FIELD) produce interlaced output with the entire output having the same field polarity (top or bottom first). * Follow, Default Top (FOLLOW_TOP_FIELD) and Follow, Default Bottom (FOLLOW_BOTTOM_FIELD) use the same field polarity as the source. Therefore, behavior depends on the input scan type, as follows. - If the source is interlaced, the output will be interlaced with the same polarity as the source (it will follow the source). The output could therefore be a mix of "top field first" and "bottom field first". - If the source is progressive, the output will be interlaced with "top field first" or "bottom field first" polarity, depending on which of the Follow options you chose.

PROGRESSIVE
TOP_FIELD
BOTTOM_FIELD
FOLLOW_TOP_FIELD
FOLLOW_BOTTOM_FIELD

H264ParControl (enum)
Using the API, enable ParFollowSource if you want the service to use the pixel aspect ratio from the input. Using the console, do this by choosing Follow source for Pixel aspect ratio.

INITIALIZE_FROM_SOURCE
H264QualityTuningLevel (enum)
Use Quality tuning level (H264QualityTuningLevel) to specify whether to use fast single-pass, high-quality singlepass, or high-quality multipass video encoding.
- SINGLE_PASS
- SINGLE_PASS_HQ
- MULTI_PASS_HQ

H264RateControlMode (enum)
Use this setting to specify whether this output has a variable bitrate (VBR) or constant bitrate (CBR).
- VBR
- CBR

H264RepeatPps (enum)
Places a PPS header on each encoded picture, even if repeated.
- DISABLED
- ENABLED

H264SceneChangeDetect (enum)
Scene change detection (inserts I-frames on scene changes).
- DISABLED
- ENABLED

H264Settings
slices
Number of slices per picture. Must be less than or equal to the number of macroblock rows for progressive pictures, and less than or equal to half the number of macroblock rows for interlaced pictures.
- Type: integer
- Required: False
- Minimum: 1
- Maximum: 32

minIInterval
Enforces separation between repeated (cadence) I-frames and I-frames inserted by Scene Change Detection. If a scene change I-frame is within I-interval frames of a cadence I-frame, the GOP is shrunk and/or stretched to the scene change I-frame. GOP stretch requires enabling lookahead as well as setting I-interval. The normal cadence resumes for the next GOP. This setting is only used when Scene Change Detect is enabled. Note: Maximum GOP stretch = GOP size + Min-I-interval - 1
- Type: integer
parNumerator

Pixel Aspect Ratio numerator.

Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

flickerAdaptiveQuantization

Type: string
Required: False

gopSizeUnits

Type: string
Required: False

hrdBufferSize

Size of buffer (HRD buffer model) in bits. For example, enter five megabits as 5000000.

Type: integer
Required: False
Minimum: 0
Maximum: 1152000000

qualityTuningLevel

Type: string
Required: False

maxBitrate

Maximum bitrate in bits/second. For example, enter five megabits per second as 5000000.

Type: integer
Required: False
Minimum: 1000
Maximum: 1152000000

bitrate

Average bitrate in bits/second. Required for VBR and CBR. For MS Smooth outputs, bitrates must be unique when rounded down to the nearest multiple of 1000.

Type: integer
Required: False
Properties

**Minimum**: 1000
**Maximum**: 1152000000

**spatialAdaptiveQuantization**
- **Type**: string
- **Required**: False

**slowPal**
- **Type**: string
- **Required**: False

**codecProfile**
- **Type**: string
- **Required**: False

**unregisteredSeiTimecode**
- **Type**: string
- **Required**: False

**softness**
Softness. Selects quantizer matrix, larger values reduce high-frequency content in the encoded image.
- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 128

**framerateControl**
- **Type**: string
- **Required**: False

**telecine**
- **Type**: string
- **Required**: False

**codecLevel**
- **Type**: string
- **Required**: False

**framerateConversionAlgorithm**
- **Type**: string
- **Required**: False
**numberReferenceFrames**

Number of reference frames to use. The encoder may use more than requested if using B-frames and/or interlaced encoding.

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 6

**temporalAdaptiveQuantization**

- **Type**: string
- **Required**: False

**repeatPps**

- **Type**: string
- **Required**: False

**hrdBufferInitialFillPercentage**

Percentage of the buffer that should initially be filled (HRD buffer model).

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 100

**framerateNumerator**

Framerate numerator - framerate is a fraction, e.g. 24000 / 1001 = 23.976 fps.

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 2147483647

**gopClosedCadence**

Frequency of closed GOPs. In streaming applications, it is recommended that this be set to 1 so a decoder joining mid-stream will receive an IDR frame as quickly as possible. Setting this value to 0 will break output segmenting.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 2147483647

**numberBFramesBetweenReferenceFrames**

Number of B-frames between reference frames.

- **Type**: integer
- **Required**: False
Properties

framerateDenominator

When you use the API for transcode jobs that use framerate conversion, specify the framerate as a fraction. For example, 24000 / 1001 = 23.976 fps. Use FramerateDenominator to specify the denominator of this fraction. In this example, use 1001 for the value of FramerateDenominator. When you use the console for transcode jobs that use framerate conversion, provide the value as a decimal number for Framerate. In this example, specify 23.976.

Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

entropyEncoding

Type: string
Required: False

fieldEncoding

Type: string
Required: False

adaptiveQuantization

Type: string
Required: False

interlaceMode

Type: string
Required: False

gopSize

GOP Length (keyframe interval) in frames or seconds. Must be greater than zero.

Type: number
Required: False
Format: float
Minimum: 0.0

gopBReference

Type: string
Required: False

syntax

Type: string
Required: False

parDenominator

Pixel Aspect Ratio denominator.

Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

sceneChangeDetect

Type: string
Required: False

parControl

Type: string
Required: False

rateControlMode

Type: string
Required: False

H264SlowPal (enum)

Enables Slow PAL rate conversion. 23.976fps and 24fps input is relabeled as 25fps, and audio is sped up correspondingly.

DISABLED
ENABLED

H264SpatialAdaptiveQuantization (enum)

Adjust quantization within each frame based on spatial variation of content complexity.

DISABLED
ENABLED

H264Syntax (enum)

Produces a bitstream compliant with SMPTE RP-2027.

DEFAULT
RP2027

H264Telecine (enum)

This field applies only if the Streams > Advanced > Framerate (framerate) field is set to 29.970. This field works with the Streams > Advanced > Preprocessors > Deinterlacer field (deinterlace_mode) and the Streams > Advanced > Interlaced Mode field (interlace_mode) to identify the scan type for the output:
Progressive, Interlaced, Hard Telecine or Soft Telecine. - Hard: produces 29.97i output from 23.976 input. - Soft: produces 23.976; the player converts this output to 29.97i.

NONE
SOFT
HARD

H264TemporalAdaptiveQuantization (enum)
Adjust quantization within each frame based on temporal variation of content complexity.

DISABLED
ENABLED

H264UnregisteredSeiTimecode (enum)
Inserts timecode for each frame as 4 bytes of an unregistered SEI message.

DISABLED
ENABLED

H265AdaptiveQuantization (enum)
Adaptive quantization. Allows intra-frame quantizers to vary to improve visual quality.

OFF
LOW
MEDIUM
HIGH
HIGHER
MAX

H265AlternateTransferFunctionSei (enum)
Enables Alternate Transfer Function SEI message for outputs using Hybrid Log Gamma (HLG) Electro-Optical Transfer Function (EOTF).

DISABLED
ENABLED

H265CodecLevel (enum)
H.265 Level.

AUTO
LEVEL_1
LEVEL_2
LEVEL_2_1
LEVEL_3
LEVEL_3_1
LEVEL_4
LEVEL_4_1
LEVEL_5
LEVEL_5_1
Properties

H265CodecProfile (enum)
Represents the Profile and Tier, per the HEVC (H.265) specification. Selections are grouped as [Profile] / [Tier], so "Main/High" represents Main Profile with High Tier. 4:2:2 profiles are only available with the HEVC 4:2:2 License.

- MAIN_MAIN
- MAIN_HIGH
- MAIN10_MAIN
- MAIN10_HIGH
- MAIN_422_8BIT_MAIN
- MAIN_422_8BIT_HIGH
- MAIN_422_10BIT_MAIN
- MAIN_422_10BIT_HIGH

H265FlickerAdaptiveQuantization (enum)
Adjust quantization within each frame to reduce flicker or 'pop' on I-frames.

- DISABLED
- ENABLED

H265FramerateControl (enum)
If you are using the console, use the Framerate setting to specify the framerate for this output. If you want to keep the same framerate as the input video, choose Follow source. If you want to do framerate conversion, choose a framerate from the dropdown list or choose Custom. The framerates shown in the dropdown list are decimal approximations of fractions. If you choose Custom, specify your framerate as a fraction. If you are creating your transcoding job specification as a JSON file without the console, use FramerateControl to specify which value the service uses for the framerate for this output. Choose INITIALIZE_FROM_SOURCE if you want the service to use the framerate from the input. Choose SPECIFIED if you want the service to use the framerate you specify in the settings FramerateNumerator and FramerateDenominator.

- INITIALIZE_FROM_SOURCE
- SPECIFIED

H265FramerateConversionAlgorithm (enum)
When set to INTERPOLATE, produces smoother motion during framerate conversion.

- DUPLICATE_DROP
- INTERPOLATE

H265GopBReference (enum)
If enable, use reference B frames for GOP structures that have B frames > 1.

- DISABLED
- ENABLED
H265GopSizeUnits (enum)

Indicates if the GOP Size in H265 is specified in frames or seconds. If seconds the system will convert the
GOP Size into a frame count at run time.

    FRAMES
    SECONDS

H265InterlaceMode (enum)

Use Interlace mode (InterlaceMode) to choose the scan line type for the output. * Top Field First
(TOP_FIELD) and Bottom Field First (BOTTOM_FIELD) produce interlaced output with the entire output
having the same field polarity (top or bottom first). * Follow, Default Top (FOLLOW_TOP_FIELD) and
Follow, Default Bottom (FOLLOW_BOTTOM_FIELD) use the same field polarity as the source. Therefore,
behavior depends on the input scan type. - If the source is interlaced, the output will be interlaced with
the same polarity as the source (it will follow the source). The output could therefore be a mix of "top
field first" and "bottom field first". - If the source is progressive, the output will be interlaced with "top
field first" or "bottom field first" polarity, depending on which of the Follow options you chose.

    PROGRESSIVE
    TOP_FIELD
    BOTTOM_FIELD
    FOLLOW_TOP_FIELD
    FOLLOW_BOTTOM_FIELD

H265ParControl (enum)

Using the API, enable ParFollowSource if you want the service to use the pixel aspect ratio from the
input. Using the console, do this by choosing Follow source for Pixel aspect ratio.

    INITIALIZE_FROM_SOURCE
    SPECIFIED

H265QualityTuningLevel (enum)

Use Quality tuning level (H265QualityTuningLevel) to specify whether to use fast single-pass, high-
quality singlepass, or high-quality multipass video encoding.

    SINGLE_PASS
    SINGLE_PASS_HQ
    MULTI_PASS_HQ

H265RateControlMode (enum)

Use this setting to specify whether this output has a variable bitrate (VBR) or constant bitrate (CBR).

    VBR
    CBR

H265SampleAdaptiveOffsetFilterMode (enum)

Specify Sample Adaptive Offset (SAO) filter strength. Adaptive mode dynamically selects best strength
based on content

    DEFAULT
H265SceneChangeDetect (enum)

Scene change detection (inserts I-frames on scene changes).

- DISABLED
- ENABLED

H265Settings

slices

Number of slices per picture. Must be less than or equal to the number of macroblock rows for progressive pictures, and less than or equal to half the number of macroblock rows for interlaced pictures.

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 32

minIInterval

Enforces separation between repeated (cadence) I-frames and I-frames inserted by Scene Change Detection. If a scene change I-frame is within I-interval frames of a cadence I-frame, the GOP is shrunk and/or stretched to the scene change I-frame. GOP stretch requires enabling lookahead as well as setting I-interval. The normal cadence resumes for the next GOP. This setting is only used when Scene Change Detect is enabled. Note: Maximum GOP stretch = GOP size + Min-I-interval - 1

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 30

parNumerator

Pixel Aspect Ratio numerator.

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 2147483647

flickerAdaptiveQuantization

- **Type**: string
- **Required**: False

gopSizeUnits

- **Type**: string
- **Required**: False
**hrdBufferSize**

Size of buffer (HRD buffer model) in bits. For example, enter five megabits as 5000000.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
- **Maximum:** 1466400000

**qualityTuningLevel**

- **Type:** string
- **Required:** False

**maxBitrate**

Maximum bitrate in bits/second.

- **Type:** integer
- **Required:** False
- **Minimum:** 1000
- **Maximum:** 1466400000

**bitrate**

Average bitrate in bits/second. Required for VBR and CBR. For MS Smooth outputs, bitrates must be unique when rounded down to the nearest multiple of 1000.

- **Type:** integer
- **Required:** False
- **Minimum:** 1000
- **Maximum:** 1466400000

**spatialAdaptiveQuantization**

- **Type:** string
- **Required:** False

**sampleAdaptiveOffsetFilterMode**

- **Type:** string
- **Required:** False

**temporalIds**

- **Type:** string
- **Required:** False

**slowPal**

- **Type:** string
- **Required:** False
tiles
  Type: string
  Required: False

writeMp4PackagingType
  Type: string
  Required: False

codecProfile
  Type: string
  Required: False

alternateTransferFunctionSei
  Type: string
  Required: False

unregisteredSeiTimecode
  Type: string
  Required: False

framerateControl
  Type: string
  Required: False

telecine
  Type: string
  Required: False

codecLevel
  Type: string
  Required: False

framerateConversionAlgorithm
  Type: string
  Required: False

numberReferenceFrames
  Number of reference frames to use. The encoder may use more than requested if using B-frames and/or interlaced encoding.
  Type: integer
  Required: False
Minimum: 1  
Maximum: 6

**temporalAdaptiveQuantization**

Type: string  
Required: False

**hrdBufferInitialFillPercentage**

Percentage of the buffer that should initially be filled (HRD buffer model).

Type: integer  
Required: False  
Minimum: 0  
Maximum: 100

**framerateNumerator**

Framerate numerator - framerate is a fraction, e.g. 24000 / 1001 = 23.976 fps.

Type: integer  
Required: False  
Minimum: 1  
Maximum: 2147483647

**gopClosedCadence**

Frequency of closed GOPs. In streaming applications, it is recommended that this be set to 1 so a decoder joining mid-stream will receive an IDR frame as quickly as possible. Setting this value to 0 will break output segmenting.

Type: integer  
Required: False  
Minimum: 0  
Maximum: 2147483647

**numberBFramesBetweenReferenceFrames**

Number of B-frames between reference frames.

Type: integer  
Required: False  
Minimum: 0  
Maximum: 7

**framerateDenominator**

Framerate denominator.

Type: integer  
Required: False  
Minimum: 1  
Maximum: 2147483647
adaptiveQuantization

Type: string
Required: False

interlaceMode

Type: string
Required: False

gopSize

GOP Length (keyframe interval) in frames or seconds. Must be greater than zero.

Type: number
Required: False
Format: float
Minimum: 0.0

parDenominator

Pixel Aspect Ratio denominator.

Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

sceneChangeDetect

Type: string
Required: False

parControl

Type: string
Required: False

rateControlMode

Type: string
Required: False

H265SlowPal (enum)

Enables Slow PAL rate conversion. 23.976fps and 24fps input is relabeled as 25fps, and audio is sped up correspondingly.
DISABLED
ENABLED

H265SpatialAdaptiveQuantization (enum)
Adjust quantization within each frame based on spatial variation of content complexity.
DISABLED
ENABLED

H265Telecine (enum)
This field applies only if the Streams > Advanced > Framerate (framerate) field is set to 29.970. This field works with the Streams > Advanced > Preprocessors > Deinterlacer field (deinterlace_mode) and the Streams > Advanced > Interlaced Mode field (interlace_mode) to identify the scan type for the output:
- Progressive, Interlaced, Hard Telecine or Soft Telecine.
- Hard: produces 29.97i output from 23.976 input.
- Soft: produces 23.976; the player converts this output to 29.97i.
NONE
SOFT
HARD

H265TemporalAdaptiveQuantization (enum)
Adjust quantization within each frame based on temporal variation of content complexity.
DISABLED
ENABLED

H265TemporalIds (enum)
Enables temporal layer identifiers in the encoded bitstream. Up to 3 layers are supported depending on GOP structure: I- and P-frames form one layer, reference B-frames can form a second layer and non-reference b-frames can form a third layer. Decoders can optionally decode only the lower temporal layers to generate a lower frame rate output. For example, given a bitstream with temporal IDs and with b-frames = 1 (i.e. IbPbPb display order), a decoder could decode all the frames for full frame rate output or only the I and P frames (lowest temporal layer) for a half frame rate output.
DISABLED
ENABLED

H265Tiles (enum)
Enable use of tiles, allowing horizontal as well as vertical subdivision of the encoded pictures.
DISABLED
ENABLED

H265UnregisteredSeiTimecode (enum)
Inserts timecode for each frame as 4 bytes of an unregistered SEI message.
DISABLED
ENABLED

H265WriteMp4PackagingType (enum)

If HVC1, output that is H.265 will be marked as HVC1 and adhere to the ISO-IECJTC1-SC29_N13798_Text_ISOIEC_FDIS_14496-15_3rd_E spec which states that parameter set NAL units will be stored in the sample headers but not in the samples directly. If HEV1, then H.265 will be marked as HEV1 and parameter set NAL units will be written into the samples.

  HVC1
  HEV1

Hdr10Metadata

redPrimaryY

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

  Type: integer
  Required: False
  Minimum: 0
  Maximum: 50000

greenPrimaryY

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

  Type: integer
  Required: False
  Minimum: 0
  Maximum: 50000

whitePointX

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

  Type: integer
  Required: False
  Minimum: 0
  Maximum: 50000

maxLuminance

Nominal maximum mastering display luminance in units of of 0.0001 candelas per square meter.

  Type: integer
  Required: False
greenPrimaryX

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

Type: integer  
Required: False  
Minimum: 0  
Maximum: 50000

whitePointY

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

Type: integer  
Required: False  
Minimum: 0  
Maximum: 50000

redPrimaryX

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

Type: integer  
Required: False  
Minimum: 0  
Maximum: 50000

bluePrimaryX

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

Type: integer  
Required: False  
Minimum: 0  
Maximum: 50000

bluePrimaryY

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

Type: integer
**Properties**

**Required**: False
**Minimum**: 0
**Maximum**: 50000

**maxFrameAverageLightLevel**
Maximum average light level of any frame in the coded video sequence, in units of candelas per square meter.

**Type**: integer
**Required**: True
**Minimum**: 0
**Maximum**: 65535

**maxContentLightLevel**
Maximum light level among all samples in the coded video sequence, in units of candelas per square meter.

**Type**: integer
**Required**: True
**Minimum**: 0
**Maximum**: 65535

**minLuminance**
Nominal minimum mastering display luminance in units of 0.0001 candelas per square meter.

**Type**: integer
**Required**: False
**Minimum**: 0
**Maximum**: 2147483647

**HlsAdMarkers (enum)**

- ELEMENTAL
- ELEMENTAL_SCTE35

**HlsAudioTrackType (enum)**

Four types of audio-only tracks are supported: Audio-Only Variant Stream The client can play back this audio-only stream instead of video in low-bandwidth scenarios. Represented as an EXT-X-STREAM-INF in the HLS manifest. Alternate Audio, Auto Select, Default Alternate rendition that the client should try to play back by default. Represented as an EXT-X-MEDIA in the HLS manifest with DEFAULT=YES, AUTOSELECT=YES Alternate Audio, Auto Select, Not Default Alternate rendition that the client may try to play back by default. Represented as an EXT-X-MEDIA in the HLS manifest with DEFAULT=NO, AUTOSELECT=YES Alternate Audio, not Auto Select Alternate rendition that the client will not try to play back by default. Represented as an EXT-X-MEDIA in the HLS manifest with DEFAULT=NO, AUTOSELECT=NO

- ALTERNATE_AUDIO_AUTO_SELECT_DEFAULT
- ALTERNATE_AUDIO_AUTO_SELECT
- ALTERNATE_AUDIO_NOT_AUTO_SELECT
HlsCaptionLanguageMapping

languageDescription
Caption language description.

Type: string
Required: False

captionChannel
Caption channel.

Type: integer
Required: False
Minimum: -2147483648
Maximum: 2147483647

languageCode

Type: string
Required: False

customLanguageCode
Specify the language for this caption channel, using the ISO 639-2 or ISO 639-3 three-letter language code

Type: string
Required: False
Pattern: ^[A-Za-z]{3}$

HlsCaptionLanguageSetting (enum)
Applies only to 608 Embedded output captions. Insert: Include CLOSED-CAPTIONS lines in the manifest. Specify at least one language in the CC1 Language Code field. One CLOSED-CAPTION line is added for each Language Code you specify. Make sure to specify the languages in the order in which they appear in the original source (if the source is embedded format) or the order of the caption selectors (if the source is other than embedded). Otherwise, languages in the manifest will not match up properly with the output captions. None: Include CLOSED-CAPTIONS=None line in the manifest. Omit: Omit any CLOSED-CAPTIONS line from the manifest.

INSERT
OMIT
NONE

HlsClientCache (enum)
When set to ENABLED, sets #EXT-X-ALLOW-CACHE:no tag, which prevents client from saving media segments for later replay.
**DISABLED**
**ENABLED**

**HlsCodecSpecification (enum)**
Specification to use (RFC-6381 or the default RFC-4281) during m3u8 playlist generation.

- RFC_6381
- RFC_4281

**HlsDirectoryStructure (enum)**
Indicates whether segments should be placed in subdirectories.

- SINGLE_DIRECTORY
- SUBDIRECTORY_PER_STREAM

**HlsEncryptionSettings**

**initializationVectorInManifest**

Type: string  
Required: False

**constantInitializationVector**
This is a 128-bit, 16-byte hex value represented by a 32-character text string. If this parameter is not set then the Initialization Vector will follow the segment number by default.

Type: string  
Required: False  
Pattern: `^[0-9a-fA-F]{32}$`

**staticKeyProvider**

Type: StaticKeyProvider (p. 341)  
Required: False

**type**

Type: string  
Required: True

**encryptionMethod**

Type: string  
Required: False

**spekeKeyProvider**

Type: SpekeKeyProvider (p. 340)
**HlsEncryptionType (enum)**

Encrypts the segments with the given encryption scheme. Leave blank to disable. Selecting 'Disabled' in the web interface also disables encryption.

AES128
SAMPLE_AES

**HlsGroupSettings**

**segmentsPerSubdirectory**

Number of segments to write to a subdirectory before starting a new one. directoryStructure must be SINGLE_DIRECTORY for this setting to have an effect.

*Type*: integer  
*Required*: False  
*Minimum*: 1  
*Maximum*: 2147483647

**streamInfResolution**

*Type*: string  
*Required*: False

**timestampDeltaMilliseconds**

Provides an extra millisecond delta offset to fine tune the timestamps.

*Type*: integer  
*Required*: False  
*Minimum*: -2147483648  
*Maximum*: 2147483647

**outputSelection**

*Type*: string  
*Required*: False

**captionLanguageMappings**

Language to be used on Caption outputs

*Type*: Array of type HlsCaptionLanguageMapping (p. 292)  
*Required*: False

**clientCache**

*Type*: string  
*Required*: False
codecSpecification
Type: string
Required: False

destination
Use Destination (Destination) to specify the S3 output location and the output filename base. Destination accepts format identifiers. If you do not specify the base filename in the URI, the service will use the filename of the input file. If your job has multiple inputs, the service uses the filename of the first input file.
Type: string
Required: False
Pattern: ^s3:\/[\/]\

timedMetadataId3Frame
Type: string
Required: False

segmentControl
Type: string
Required: False

timedMetadataId3Period
Timed Metadata interval in seconds.
Type: integer
Required: True
Minimum: -2147483648
Maximum: 2147483647

captionLanguageSetting
Type: string
Required: False

minSegmentLength
When set, Minimum Segment Size is enforced by looking ahead and back within the specified range for a nearby avail and extending the segment size if needed.
Type: integer
Required: True
Minimum: 0
Maximum: 2147483647

directoryStructure
Type: string
Required: False
**programDateTime**

_Type:_ string  
_Required:_ False

**baseUrl**

A partial URI prefix that will be prepended to each output in the media .m3u8 file. Can be used if base manifest is delivered from a different URL than the main .m3u8 file.

_Type:_ string  
_Required:_ False

**encryption**

DRM settings.

_Type:_ HlsEncryptionSettings (p. 293)  
_Required:_ False

**adMarkers**

Choose one or more ad marker types to pass SCTE35 signals through to this group of Apple HLS outputs.

_Type:_ Array of type string  
_Required:_ False

**programDateTimePeriod**

Period of insertion of EXT-X-PROGRAM-DATE-TIME entry, in seconds.

_Type:_ integer  
_Required:_ False  
_Minimum:_ 0  
_Maximum:_ 3600

**manifestCompression**

_Type:_ string  
_Required:_ False

**segmentLength**

Length of MPEG-2 Transport Stream segments to create (in seconds). Note that segments will end on the next keyframe after this number of seconds, so actual segment length may be longer.

_Type:_ integer  
_Required:_ True  
_Minimum:_ 1  
_Maximum:_ 2147483647

**manifestDurationFormat**

_Type:_ string
Properties

Required: False

HlsIFrameOnlyManifest (enum)
When set to INCLUDE, writes I-Frame Only Manifest in addition to the HLS manifest

  INCLUDE
  EXCLUDE

HlsInitializationVectorInManifest (enum)
The Initialization Vector is a 128-bit number used in conjunction with the key for encrypting blocks. If set to INCLUDE, Initialization Vector is listed in the manifest. Otherwise Initialization Vector is not in the manifest.

  INCLUDE
  EXCLUDE

HlsKeyProviderType (enum)
Indicates which type of key provider is used for encryption.

  SPEKE
  STATIC_KEY

HlsManifestCompression (enum)
When set to GZIP, compresses HLS playlist.

  GZIP
  NONE

HlsManifestDurationFormat (enum)
Indicates whether the output manifest should use floating point values for segment duration.

  FLOATING_POINT
  INTEGER

HlsOutputSelection (enum)
Indicates whether the .m3u8 manifest file should be generated for this HLS output group.

  MANIFESTS_AND_SEGMENTS
  SEGMENTS_ONLY

HlsProgramDateTime (enum)
Includes or excludes EXT-X-PROGRAM-DATE-TIME tag in .m3u8 manifest files. The value is calculated as follows: either the program date and time are initialized using the input timecode source, or the time is initialized using the input timecode source and the date is initialized using the timestamp_offset.

  INCLUDE
  EXCLUDE
**HlsSegmentControl (enum)**

When set to SINGLE_FILE, emits program as a single media resource (.ts) file, uses #EXT-X-BYTERANGE tags to index segment for playback.

- SINGLE_FILE
- SEGMENTED_FILES

**HlsSettings**

**iFrameOnlyManifest**

- Type: string
- Required: False

**audioRenditionSets**

List all the audio groups that are used with the video output stream. Input all the audio GROUP-IDs that are associated to the video, separate by ','. 

- Type: string
- Required: False

**audioTrackType**

- Type: string
- Required: False

**segmentModifier**

String concatenated to end of segment filenames. Accepts "Format Identifiers":#format_identifier_parameters.

- Type: string
- Required: False

**audioGroupId**

Specifies the group to which the audio Rendition belongs.

- Type: string
- Required: False

**HlsStreamInfResolution (enum)**

Include or exclude RESOLUTION attribute for video in EXT-X-STREAM-INF tag of variant manifest.

- INCLUDE
- EXCLUDE

**HlsTimedMetadataId3Frame (enum)**

Indicates ID3 frame that has the timecode.

- NONE
### Id3Insertion

**id3**

Use ID3 tag (id3) to provide a tag value in base64-encode format.

- **Type:** string
- **Required:** True
- **Pattern:** `^[A-Za-z0-9+/=]{0,2}$`

#### timecode

Provide a Timecode (TimeCode) in HH:MM:SS:FF or HH:MM:SS;FF format.

- **Type:** string
- **Required:** True
- **Format:** timecode
- **Pattern:** `^[01][0-9]|2[0-4]):[0-5][0-9]:[0-5][0-9];;;[0-9]{2}$`

### ImageInserter

**insertableImages**

Image to insert. Must be 32 bit windows BMP, PNG, or TGA file. Must not be larger than the output frames.

- **Type:** Array of type InsertableImage (p. 302)
- **Required:** True

### InputClipping

**startTimecode**

Set Start timecode (StartTimecode) to the beginning of the portion of the input you are clipping. The frame corresponding to the Start timecode value is included in the clip. Start timecode or End timecode may be left blank, but not both. Use the format HH:MM:SS:FF or HH:MM:SS;FF, where HH is the hour, MM is the minute, SS is the second, and FF is the frame number. When choosing this value, take into account your setting for Input timecode source. For example, if you have embedded timecodes that start at 01:00:00:00 and you want your clip to begin five minutes into the video, use 01:05:00:00.

- **Type:** string
- **Required:** False
- **Format:** timecode
- **Pattern:** `^[01][0-9]|2[0-4]):[0-5][0-9]:[0-5][0-9];;;[0-9]{2}$`

**endTimecode**

Set End timecode (EndTimecode) to the end of the portion of the input you are clipping. The frame corresponding to the End timecode value is included in the clip. Start timecode or End timecode may be left blank, but not both. Use the format HH:MM:SS:FF or HH:MM:SS;FF, where HH is the hour, MM is the minute, SS is the second, and FF is the frame number. When choosing this value, take into account
your setting for timecode source under input settings (InputTimecodeSource). For example, if you have embedded timecodes that start at 01:00:00:00 and you want your clip to end six minutes into the video, use 01:06:00:00.

Type: string  
Required: False  
Format: timecode  
Pattern: ^([01][0-9]|2[0-4]):[0-5][0-9]:[0-5][0-9]:;[0-9]{2}$

InputDeblockFilter (enum)
Enable Deblock (InputDeblockFilter) to produce smoother motion in the output. Default is disabled. Only manually controllable for MPEG2 and uncompressed video inputs.

ENABLED  
DISABLED

InputDenoiseFilter (enum)
Enable Denoise (InputDenoiseFilter) to filter noise from the input. Default is disabled. Only applicable to MPEG2, H.264, H.265, and uncompressed video inputs.

ENABLED  
DISABLED

InputFilterEnable (enum)
Use Filter enable (InputFilterEnable) to specify how the transcoding service applies the denoise and deblock filters. You must also enable the filters separately, with Denoise (InputDenoiseFilter) and Deblock (InputDeblockFilter). * Auto - The transcoding service determines whether to apply filtering, depending on input type and quality. * Disable - The input is not filtered. This is true even if you use the API to enable them in (InputDeblockFilter) and (InputDeblockFilter). * Force - The input is filtered regardless of input type.

AUTO  
DISABLE  
FORCE

InputPsiControl (enum)
Set PSI control (InputPsiControl) for transport stream inputs to specify which data the demux process to scans. * Ignore PSI - Scan all PIDs for audio and video. * Use PSI - Scan only PSI data.

IGNORE_PSI  
USE_PSI

InputTemplate

audioSelectors
Use Audio selectors (AudioSelectors) to specify a track or set of tracks from the input that you will use in your outputs. You can use mutiple Audio selectors per input.

Type: object  
Required: False
audioSelectorGroups

Specifies set of audio selectors within an input to combine. An input may have multiple audio selector
groups. See "Audio Selector Group" for more information.

- **Type**: object
- **Required**: False

filterEnable

- **Type**: string
- **Required**: False

deblockFilter

- **Type**: string
- **Required**: False

videoSelector

- **Type**: VideoSelector
- **Required**: False

filterStrength

Use Filter strength (FilterStrength) to adjust the magnitude the input filter settings (Deblock and
Denoise). The range is -5 to 5. Default is 0.

- **Type**: integer
- **Required**: False
- **Minimum**: -5
- **Maximum**: 5

programNumber

Use Program (programNumber) to select a specific program from within a multi-program transport
stream. Note that Quad 4K is not currently supported. Default is the first program within the transport
stream. If the program you specify doesn't exist, the transcoding service will use this default.

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 2147483647

timecodeSource

- **Type**: string
- **Required**: False

denoiseFilter

- **Type**: string
Required: False

captionSelectors

Use Captions selectors (CaptionSelectors) to specify the captions data from the input that you will use in your outputs. You can use multiple captions selectors per input.

Type: object
Required: False

inputClippings

(InputClippings) contains sets of start and end times that together specify a portion of the input to be used in the outputs. If you provide only a start time, the clip will be the entire input from that point to the end. If you provide only an end time, it will be the entire input up to that point. When you specify more than one input clip, the transcoding service creates the job outputs by stringing the clips together in the order you specify them.

Type: Array of type InputClipping (p. 299)
Required: False

psiControl

Type: string
Required: False

InputTimecodeSource (enum)

Timecode source under input settings (InputTimecodeSource) only affects the behavior of features that apply to a single input at a time, such as input clipping and synchronizing some captions formats. Use this setting to specify whether the service counts frames by timecodes embedded in the video (EMBEDDED) or by starting the first frame at zero (ZEROBASED). In both cases, the timecode format is HH:MM:SS:FF or HH:MM:SS;FF, where FF is the frame number. Only set this to EMBEDDED if your source video has embedded timecodes.

EMBEDDED
ZEROBASED
SPECIFIEDSTART

InsertableImage

duration

Use Duration (Duration) to set the time, in milliseconds, for the image to remain on the output video.

Type: integer
Required: False
Minimum: -2147483648
Maximum: 2147483647

fadeOut

Use Fade out (FadeOut) to set the length, in milliseconds, of the inserted image fade out. If you don’t specify a value for Fade out, the image will disappear abruptly at the end of the inserted image duration.
### Properties

**imageY**

Use Top (ImageY) to set the distance, in pixels, between the inserted image and the top edge of the video frame. Required for BMP, PNG and TGA input.

- **Type**: integer
- **Required**: True
- **Minimum**: -2147483648
- **Maximum**: 2147483647

**fadeIn**

Use Fade in (FadeIn) to set the length, in milliseconds, of the inserted image fade in. If you don't specify a value for Fade in, the image will appear abruptly at the Start time.

- **Type**: integer
- **Required**: False
- **Minimum**: -2147483648
- **Maximum**: 2147483647

**imageX**

Use Left (ImageX) to set the distance, in pixels, between the inserted image and the left edge of the frame. Required for BMP, PNG and TGA input.

- **Type**: integer
- **Required**: True
- **Minimum**: -2147483648
- **Maximum**: 2147483647

**width**

Specify the Width (Width) of the inserted image. Use a value that is less than or equal to the video resolution width. Leave this setting blank to use the native width of the image.

- **Type**: integer
- **Required**: False
- **Minimum**: -2147483648
- **Maximum**: 2147483647

**startTime**

Use Start time (startTime) to specify the video timecode when the image is inserted in the output. This must be in timecode (HH:MM:SS:FF or HH:MM:SS;FF) format.

- **Type**: string
- **Required**: False
- **Pattern**: `(^[0-1]?[0-9]|2[0-3]):([0-5][0-9]|0):([0-5][0-9]|0)`
opacity

Use Opacity (Opacity) to specify how much of the underlying video shows through the inserted image. 0 is transparent and 100 is fully opaque. Default is 50.

Type: integer
Required: True
Minimum: 0
Maximum: 100

layer

Use Layer (Layer) to specify how overlapping inserted images appear. Images with higher values of layer appear on top of images with lower values of layer.

Type: integer
Required: True
Minimum: 0
Maximum: 99

height

Specify the Height (Height) of the inserted image. Use a value that is less than or equal to the video resolution height. Leave this setting blank to use the native height of the image.

Type: integer
Required: False
Minimum: -2147483648
Maximum: 2147483647

imageInserterInput

Use Image location (imageInserterInput) to specify the Amazon S3 location of the image to be inserted into the output. Use a 32 bit BMP, PNG, or TGA file that fits inside the video frame.

Type: string
Required: True

Pattern: ^s3:/

JobTemplate

createdAt

The timestamp in epoch seconds for Job template creation.

Type: string
Required: False
Format: date-time

lastUpdated

The timestamp in epoch seconds when the Job template was last updated.
Type: string
Required: False
Format: date-time

settings
Type: JobTemplateSettings (p. 306)
Required: True

name
A name you create for each job template. Each name must be unique within your account.

Type: string
Required: True

description
An optional description you create for each job template.

Type: string
Required: False

arn
An identifier for this resource that is unique within all of AWS.

Type: string
Required: False

category
An optional category you create to organize your job templates.

Type: string
Required: False

type
A job template can be of two types: system or custom. System or built-in job templates can't be modified or deleted by the user.

Type: string
Required: False

queue
Optional. The queue that jobs created from this template are assigned to. If you don't specify this, jobs will go to the default queue.

Type: string
Required: False
Properties

JobTemplateSettings

timecodeConfig

Contains settings used to acquire and adjust timecode information from inputs.

Type: TimecodeConfig (p. 343)
Required: False

adAvailOffset

When specified, this offset (in milliseconds) is added to the input Ad Avail PTS time.

Type: integer
Required: False
Minimum: -1000
Maximum: 1000

outputGroups

(OutputGroups) contains one group of settings for each set of outputs that share a common package type. All unpackaged files (MPEG-4, MPEG-2 TS, Quicktime, MXF, and no container) are grouped in a single output group as well. Required in (OutputGroups) is a group of settings that apply to the whole group. This required object depends on the value you set for (Type) under (OutputGroups)>(OutputGroupSettings). Type, settings object pairs are as follows. *
FILE_GROUP_SETTINGS, FileGroupSettings *
HLS_GROUP_SETTINGS, HlsGroupSettings *
DASH_ISO_GROUP_SETTINGS, DashIsoGroupSettings *
MS_SMOOTH_GROUP_SETTINGS, MsSmoothGroupSettings *
CMAF_GROUP_SETTINGS, CmafGroupSettings

Type: Array of type OutputGroup (p. 334)
Required: True

nielsenConfiguration

Type: NielsenConfiguration (p. 331)
Required: False

inputs

Use Inputs (inputs) to define the source file used in the transcode job. There can only be one input in a job template. Using the API, you can include multiple inputs when referencing a job template.

Type: Array of type InputTemplate (p. 300)
Required: False

timedMetadataInsertion

Type: TimedMetadataInsertion (p. 344)
Required: False

availBlanking

Settings for ad avail blanking. Video can be blanked or overlaid with an image, and audio muted during SCTE-35 triggered ad avails.
**Type:** AvailBlanking (p. 241)
**Required:** False

**LanguageCode (enum)**


ENG
SPA
FRA
DEU
GER
ZHO
ARA
HIN
JPN
RUS
POR
ITA
URD
VIE
KOR
PAN
ABK
AAR
AFR
AKA
SQI
AMH
ARG
HYE
ASM
AVA
AVE
AYM
AZE
BAM
BAK
EUS
BEL
BEN
BIH
BIS
BOS
BRE
BUL
MYA
CAT
KHM
CHA
CHE
NYA
CHU
CHV
COR
COS
CRE
HRV
CES
DAN
DIV
NLD
DZO
ENM
EPO
EST
EWE
FAO
FIJ
FIN
FRM
FUL
GLA
GLG
LUG
KAT
ELL
GRN
GUJ
HAT
HAU
HEB
HER
HMO
HUN
ISL
IDO
IBO
IND
INA
ILE
IKU
IPK
GLE
JAV
KAL
KAN
KAU
KAS
KAZ
KIK
KIN
KIR
KOM
KON
KUA
KUR
LAO
LAT
<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAV</td>
<td></td>
</tr>
<tr>
<td>LIM</td>
<td></td>
</tr>
<tr>
<td>LIN</td>
<td></td>
</tr>
<tr>
<td>LIT</td>
<td></td>
</tr>
<tr>
<td>LUB</td>
<td></td>
</tr>
<tr>
<td>LTZ</td>
<td></td>
</tr>
<tr>
<td>MKD</td>
<td></td>
</tr>
<tr>
<td>MLG</td>
<td></td>
</tr>
<tr>
<td>MSA</td>
<td></td>
</tr>
<tr>
<td>MAL</td>
<td></td>
</tr>
<tr>
<td>MLT</td>
<td></td>
</tr>
<tr>
<td>GLV</td>
<td></td>
</tr>
<tr>
<td>MRI</td>
<td></td>
</tr>
<tr>
<td>MAR</td>
<td></td>
</tr>
<tr>
<td>MAH</td>
<td></td>
</tr>
<tr>
<td>MON</td>
<td></td>
</tr>
<tr>
<td>NAU</td>
<td></td>
</tr>
<tr>
<td>NAV</td>
<td></td>
</tr>
<tr>
<td>NDE</td>
<td></td>
</tr>
<tr>
<td>NBL</td>
<td></td>
</tr>
<tr>
<td>NDO</td>
<td></td>
</tr>
<tr>
<td>NEP</td>
<td></td>
</tr>
<tr>
<td>SME</td>
<td></td>
</tr>
<tr>
<td>NOR</td>
<td></td>
</tr>
<tr>
<td>NOB</td>
<td></td>
</tr>
<tr>
<td>NNO</td>
<td></td>
</tr>
<tr>
<td>OCI</td>
<td></td>
</tr>
<tr>
<td>OJI</td>
<td></td>
</tr>
<tr>
<td>ORI</td>
<td></td>
</tr>
<tr>
<td>ORM</td>
<td></td>
</tr>
<tr>
<td>OSS</td>
<td></td>
</tr>
<tr>
<td>PLI</td>
<td></td>
</tr>
<tr>
<td>FAS</td>
<td></td>
</tr>
<tr>
<td>POL</td>
<td></td>
</tr>
<tr>
<td>PUS</td>
<td></td>
</tr>
<tr>
<td>QUE</td>
<td></td>
</tr>
<tr>
<td>QAA</td>
<td></td>
</tr>
<tr>
<td>RON</td>
<td></td>
</tr>
<tr>
<td>ROH</td>
<td></td>
</tr>
<tr>
<td>RUN</td>
<td></td>
</tr>
<tr>
<td>SMO</td>
<td></td>
</tr>
<tr>
<td>SAG</td>
<td></td>
</tr>
<tr>
<td>SAN</td>
<td></td>
</tr>
<tr>
<td>SRD</td>
<td></td>
</tr>
<tr>
<td>SRB</td>
<td></td>
</tr>
<tr>
<td>SNA</td>
<td></td>
</tr>
<tr>
<td>III</td>
<td></td>
</tr>
<tr>
<td>SND</td>
<td></td>
</tr>
<tr>
<td>SIN</td>
<td></td>
</tr>
<tr>
<td>SLK</td>
<td></td>
</tr>
<tr>
<td>SLV</td>
<td></td>
</tr>
<tr>
<td>SOM</td>
<td></td>
</tr>
<tr>
<td>SOT</td>
<td></td>
</tr>
<tr>
<td>SUN</td>
<td></td>
</tr>
<tr>
<td>SWA</td>
<td></td>
</tr>
</tbody>
</table>
SSW
SWE
TGL
TAH
TGK
TAM
TAT
TEL
THA
BOD
TIR
TON
TSO
TSN
TUR
TUK
TWI
UIG
UKR
UZB
VEN
VOL
WLN
CYM
FRY
WOL
XHO
VID
YOR
ZHA
ZUL
ORJ
QPC
TNG

M2tsAudioBufferModel (enum)

Selects between the DVB and ATSC buffer models for Dolby Digital audio.

DVB
ATSC

M2tsBufferModel (enum)

Controls what buffer model to use for accurate interleaving. If set to MULTIPLEX, use multiplex buffer model. If set to NONE, this can lead to lower latency, but low-memory devices may not be able to play back the stream without interruptions.

MULTIPLEX
NONE

M2tsEbpAudioInterval (enum)

When set to VIDEO_AND_FIXED_INTERVALS, audio EBP markers will be added to partitions 3 and 4. The interval between these additional markers will be fixed, and will be slightly shorter than the video
EBP marker interval. When set to VIDEO_INTERVAL, these additional markers will not be inserted. Only applicable when EBP segmentation markers are is selected (segmentationMarkers is EBP or EBP_LEGACY).

- VIDEO_AND_FIXED_INTERVALS
- VIDEO_INTERVAL

**M2tsEbpPlacement (enum)**

Selects which PIDs to place EBP markers on. They can either be placed only on the video PID, or on both the video PID and all audio PIDs. Only applicable when EBP segmentation markers are is selected (segmentationMarkers is EBP or EBP_LEGACY).

- VIDEO_AND_AUDIO_PIDS
- VIDEO_PID

**M2tsEsRateInPes (enum)**

Controls whether to include the ES Rate field in the PES header.

- INCLUDE
- EXCLUDE

**M2tsNielsenId3 (enum)**

If INSERT, Nielsen inaudible tones for media tracking will be detected in the input audio and an equivalent ID3 tag will be inserted in the output.

- INSERT
- NONE

**M2tsPcrControl (enum)**

When set to PCR_EVERY_PES_PACKET, a Program Clock Reference value is inserted for every Packetized Elementary Stream (PES) header. This is effective only when the PCR PID is the same as the video or audio elementary stream.

- PCR_EVERY_PES_PACKET
- CONFIGURED_PCR_PERIOD

**M2tsRateMode (enum)**

When set to CBR, inserts null packets into transport stream to fill specified bitrate. When set to VBR, the bitrate setting acts as the maximum bitrate, but the output will not be padded up to that bitrate.

- VBR
- CBR

**M2tsScte35Source (enum)**

Enables SCTE-35 passthrough (scte35Source) to pass any SCTE-35 signals from input to output.

- PASSTHROUGH
- NONE
**M2tsSegmentationMarkers (enum)**

Inserts segmentation markers at each segmentation_time period. rai_segstart sets the Random Access Indicator bit in the adaptation field. rai_adapt sets the RAI bit and adds the current timecode in the private data bytes. psi_segstart inserts PAT and PMT tables at the start of segments. ebp adds Encoder Boundary Point information to the adaptation field as per OpenCable specification OC-SP-EBP-I01-130118. ebp_legacy adds Encoder Boundary Point information to the adaptation field using a legacy proprietary format.

- NONE
- RAI_SEGSTART
- RAI_ADAPT
- PSI_SEGSTART
- EBP
- EBP_LEGACY

**M2tsSegmentationStyle (enum)**

The segmentation style parameter controls how segmentation markers are inserted into the transport stream. With avails, it is possible that segments may be truncated, which can influence where future segmentation markers are inserted. When a segmentation style of "reset_cadence" is selected and a segment is truncated due to an avail, we will reset the segmentation cadence. This means the subsequent segment will have a duration of $segmentation_time seconds. When a segmentation style of "maintain_cadence" is selected and a segment is truncated due to an avail, we will not reset the segmentation cadence. This means the subsequent segment will likely be truncated as well. However, all segments after that will have a duration of $segmentation_time seconds. Note that EBP lookahead is a slight exception to this rule.

- MAINTAIN_CADENCE
- RESET_CADENCE

**M2tsSettings**

**dvbTeletextPid**

Packet Identifier (PID) for input source DVB Teletext data to this output.

- **Type**: integer
- **Required**: False
- **Minimum**: 32
- **Maximum**: 8182

**nielsenId3**

- **Type**: string
- **Required**: False

**bitrate**

The output bitrate of the transport stream in bits per second. Setting to 0 lets the muxer automatically determine the appropriate bitrate. Other common values are 3750000, 7500000, and 15000000.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
Maximum: 2147483647

segmentationTime
The length in seconds of each segment. Required unless markers is set to _none_.

  Type: number
  Required: False
  Format: float
  Minimum: 0.0

audioPids
Packet Identifier (PID) of the elementary audio stream(s) in the transport stream. Multiple values are accepted, and can be entered in ranges and/or by comma separation.

  Type: Array of type integer
  Required: False

rateMode

  Type: string
  Required: False

maxPcrInterval
Maximum time in milliseconds between Program Clock References (PCRs) inserted into the transport stream.

  Type: integer
  Required: False
  Minimum: 0
  Maximum: 500

audioFramesPerPes
The number of audio frames to insert for each PES packet.

  Type: integer
  Required: False
  Minimum: 0
  Maximum: 2147483647

ebpAudioInterval

  Type: string
  Required: False

fragmentTime
The length in seconds of each fragment. Only used with EBP markers.

  Type: number
  Required: False
Format: float
Minimum: 0.0

scte35Pid
Packet Identifier (PID) of the SCTE-35 stream in the transport stream.

Type: integer
Required: False
Minimum: 32
Maximum: 8182

pmtInterval
The number of milliseconds between instances of this table in the output transport stream.

Type: integer
Required: False
Minimum: 0
Maximum: 1000

privateMetadataPid
Packet Identifier (PID) of the private metadata stream in the transport stream.

Type: integer
Required: False
Minimum: 32
Maximum: 8182

segmentationStyle

Type: string
Required: False

audioBufferModel

Type: string
Required: False

programNumber
The value of the program number field in the Program Map Table.

Type: integer
Required: False
Minimum: 0
Maximum: 65535

dvbNitSettings

Type: DvbNitSettings (p. 258)
Required: False
timedMetadataPid
Packet Identifier (PID) of the timed metadata stream in the transport stream.

- **Type**: integer
- **Required**: False
- **Minimum**: 32
- **Maximum**: 8182

scte35Source

- **Type**: string
- **Required**: False

pmtPID
Packet Identifier (PID) for the Program Map Table (PMT) in the transport stream.

- **Type**: integer
- **Required**: False
- **Minimum**: 32
- **Maximum**: 8182

bufferModel

- **Type**: string
- **Required**: False

ebpPlacement

- **Type**: string
- **Required**: False

dvbSdtSettings

- **Type**: DvbSdtSettings (p. 258)
- **Required**: False

nullPacketBitrate
Value in bits per second of extra null packets to insert into the transport stream. This can be used if a downstream encryption system requires periodic null packets.

- **Type**: number
- **Required**: False
- **Format**: float
- **Minimum**: 0.0

pcrPid
Packet Identifier (PID) of the Program Clock Reference (PCR) in the transport stream. When no value is given, the encoder will assign the same value as the Video PID.

- **Type**: integer
Properties

Required: False
Minimum: 32
Maximum: 8182

**minEbpInterval**

When set, enforces that Encoder Boundary Points do not come within the specified time interval of each other by looking ahead at input video. If another EBP is going to come in within the specified time interval, the current EBP is not emitted, and the segment is "stretched" to the next marker. The lookahead value does not add latency to the system. The Live Event must be configured elsewhere to create sufficient latency to make the lookahead accurate.

Type: integer
Required: False
Minimum: 0
Maximum: 10000

**transportStreamId**

The value of the transport stream ID field in the Program Map Table.

Type: integer
Required: False
Minimum: 0
Maximum: 65535

**videoPid**

Packet Identifier (PID) of the elementary video stream in the transport stream.

Type: integer
Required: False
Minimum: 32
Maximum: 8182

**pcrControl**

Type: string
Required: False

**esRateInPes**

Type: string
Required: False

**segmentationMarkers**

Type: string
Required: False

**dvbTdtSettings**

Type: DvbTdtSettings (p. 263)
Required: False

**patInterval**

The number of milliseconds between instances of this table in the output transport stream.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
- **Maximum:** 1000

**dvbSubPids**

Packet Identifier (PID) for input source DVB Subtitle data to this output. Multiple values are accepted, and can be entered in ranges and/or by comma separation.

- **Type:** Array of type integer
- **Required:** False

**M3u8NielsenId3 (enum)**

If INSERT, Nielsen inaudible tones for media tracking will be detected in the input audio and an equivalent ID3 tag will be inserted in the output.

- **INSERT**
- **NONE**

**M3u8PcrControl (enum)**

When set to PCR_EVERY_PES_PACKET a Program Clock Reference value is inserted for every Packetized Elementary Stream (PES) header. This parameter is effective only when the PCR PID is the same as the video or audio elementary stream.

- **PCR_EVERY_PES_PACKET**
- **CONFIGURED_PCR_PERIOD**

**M3u8Scte35Source (enum)**

Enables SCTE-35 passthrough (scte35Source) to pass any SCTE-35 signals from input to output.

- **PASSTHROUGH**
- **NONE**

**M3u8Settings**

**pmtPid**

Packet Identifier (PID) for the Program Map Table (PMT) in the transport stream.

- **Type:** integer
- **Required:** False
- **Minimum:** 32
- **Maximum:** 8182
nielsenId3
Type: string
Required: False

pcrPid
Packet Identifier (PID) of the Program Clock Reference (PCR) in the transport stream. When no value is given, the encoder will assign the same value as the Video PID.
Type: integer
Required: False
Minimum: 32
Maximum: 8182

audioPids
Packet Identifier (PID) of the elementary audio stream(s) in the transport stream. Multiple values are accepted, and can be entered in ranges and/or by comma separation.
Type: Array of type integer
Required: False

audioFramesPerPes
The number of audio frames to insert for each PES packet.
Type: integer
Required: False
Minimum: 0
Maximum: 2147483647

scte35Pid
Packet Identifier (PID) of the SCTE-35 stream in the transport stream.
Type: integer
Required: False
Minimum: 32
Maximum: 8182

transportStreamId
The value of the transport stream ID field in the Program Map Table.
Type: integer
Required: False
Minimum: 0
Maximum: 65535

pcrControl
Type: string
Required: False

videoPid
Packet Identifier (PID) of the elementary video stream in the transport stream.

Type: integer
Required: False
Minimum: 32
Maximum: 8182

privateMetadataPid
Packet Identifier (PID) of the private metadata stream in the transport stream.

Type: integer
Required: False
Minimum: 32
Maximum: 8182

pmtInterval
The number of milliseconds between instances of this table in the output transport stream.

Type: integer
Required: False
Minimum: 0
Maximum: 1000

patInterval
The number of milliseconds between instances of this table in the output transport stream.

Type: integer
Required: False
Minimum: 0
Maximum: 1000

programNumber
The value of the program number field in the Program Map Table.

Type: integer
Required: False
Minimum: 0
Maximum: 65535

timedMetadataPid
Packet Identifier (PID) of the timed metadata stream in the transport stream.

Type: integer
Required: False
Properties

Minimum: 32
Maximum: 8182

timedMetadata
Type: string
Required: False

scte35Source
Type: string
Required: False

MovClapAtom (enum)
When enabled, include 'clap' atom if appropriate for the video output settings.

INCLUDE
EXCLUDE

MovCslgAtom (enum)
When enabled, file composition times will start at zero, composition times in the 'ctts' (composition time to sample) box for B-frames will be negative, and a 'cslg' (composition shift least greatest) box will be included per 14496-1 amendment 1. This improves compatibility with Apple players and tools.

INCLUDE
EXCLUDE

MovMpeg2FourCCControl (enum)
When set to XDCAM, writes MPEG2 video streams into the QuickTime file using XDCAM fourcc codes. This increases compatibility with Apple editors and players, but may decrease compatibility with other players. Only applicable when the video codec is MPEG2.

XDCAM
MPEG

MovPaddingControl (enum)
If set to OMNEON, inserts Omneon-compatible padding

OMNEON
NONE

MovReference (enum)
A value of 'external' creates separate media files and the wrapper file (.mov) contains references to these media files. A value of 'self_contained' creates only a wrapper (.mov) file and this file contains all of the media.

SELF_CONTAINED
EXTERNAL
MovSettings

reference
Type: string
Required: False

paddingControl
Type: string
Required: False

mpeg2FourCCControl
Type: string
Required: False

cslgAtom
Type: string
Required: False

clapAtom
Type: string
Required: False

Mp2Settings

channels
Set Channels to specify the number of channels in this output audio track. Choosing Mono in the console will give you 1 output channel; choosing Stereo will give you 2. In the API, valid values are 1 and 2.

Type: integer
Required: False
Minimum: 1
Maximum: 2

bitrate
Average bitrate in bits/second.

Type: integer
Required: False
Minimum: 32000
Maximum: 384000

sampleRate
Sample rate in hz.
**Properties**

**Type**
- Type: integer
- Required: False
- Minimum: 32000
- Maximum: 48000

**Mp4CslgAtom (enum)**

When enabled, file composition times will start at zero, composition times in the 'ctts' (composition time to sample) box for B-frames will be negative, and a 'cslg' (composition shift least greatest) box will be included per 14496-1 amendment 1. This improves compatibility with Apple players and tools.

- INCLUDE
- EXCLUDE

**Mp4FreeSpaceBox (enum)**

Inserts a free-space box immediately after the moov box.

- INCLUDE
- EXCLUDE

**Mp4MoovPlacement (enum)**

If set to PROGRESSIVE_DOWNLOAD, the MOOV atom is relocated to the beginning of the archive as required for progressive downloading. Otherwise it is placed normally at the end.

- PROGRESSIVE_DOWNLOAD
- NORMAL

**Mp4Settings**

**mp4MajorBrand**

Overrides the "Major Brand" field in the output file. Usually not necessary to specify.

- Type: string
- Required: False

**moovPlacement**

- Type: string
- Required: False

**cslgAtom**

- Type: string
- Required: False

**freeSpaceBox**

- Type: string
- Required: False
**Mpeg2AdaptiveQuantization (enum)**

Adaptive quantization. Allows intra-frame quantizers to vary to improve visual quality.

- OFF
- LOW
- MEDIUM
- HIGH

**Mpeg2CodecLevel (enum)**

Use Level (Mpeg2CodecLevel) to set the MPEG-2 level for the video output.

- AUTO
- LOW
- MAIN
- HIGH1440
- HIGH

**Mpeg2CodecProfile (enum)**

Use Profile (Mpeg2CodecProfile) to set the MPEG-2 profile for the video output.

- MAIN
- PROFILE_422

**Mpeg2FramerateControl (enum)**

If you are using the console, use the Framerate setting to specify the framerate for this output. If you want to keep the same framerate as the input video, choose Follow source. If you want to do framerate conversion, choose a framerate from the dropdown list or choose Custom. The framerates shown in the dropdown list are decimal approximations of fractions. If you choose Custom, specify your framerate as a fraction. If you are creating your transcoding job specification as a JSON file without the console, use FramerateControl to specify which value the service uses for the framerate for this output. Choose INITIALIZE_FROM_SOURCE if you want the service to use the framerate from the input. Choose SPECIFIED if you want the service to use the framerate you specify in the settings FramerateNumerator and FramerateDenominator.

- INITIALIZE_FROM_SOURCE
- SPECIFIED

**Mpeg2FramerateConversionAlgorithm (enum)**

When set to INTERPOLATE, produces smoother motion during framerate conversion.

- DUPLICATE_DROP
- INTERPOLATE

**Mpeg2GopSizeUnits (enum)**

Indicates if the GOP Size in MPEG2 is specified in frames or seconds. If seconds the system will convert the GOP Size into a frame count at run time.

- FRAMES
SECONDS

Mpeg2InterlaceMode (enum)

Use Interlace mode (InterlaceMode) to choose the scan line type for the output. * Top Field First (TOP_FIELD) and Bottom Field First (BOTTOM_FIELD) produce interlaced output with the entire output having the same field polarity (top or bottom first). * Follow, Default Top (FOLLOW_TOP_FIELD) and Follow, Default Bottom (FOLLOW_BOTTOM_FIELD) use the same field polarity as the source. Therefore, behavior depends on the input scan type. - If the source is interlaced, the output will be interlaced with the same polarity as the source (it will follow the source). The output could therefore be a mix of “top field first” and “bottom field first”. - If the source is progressive, the output will be interlaced with “top field first” or “bottom field first” polarity, depending on which of the Follow options you chose.

PROGRESSIVE
TOP_FIELD
BOTTOM_FIELD
FOLLOW_TOP_FIELD
FOLLOW_BOTTOM_FIELD

Mpeg2IntraDcPrecision (enum)

Use Intra DC precision (Mpeg2IntraDcPrecision) to set quantization precision for intra-block DC coefficients. If you choose the value auto, the service will automatically select the precision based on the per-frame compression ratio.

AUTO
INTRA_DC_PRECISION_8
INTRA_DC_PRECISION_9
INTRA_DC_PRECISION_10
INTRA_DC_PRECISION_11

Mpeg2ParControl (enum)

Using the API, enable ParFollowSource if you want the service to use the pixel aspect ratio from the input. Using the console, do this by choosing Follow source for Pixel aspect ratio.

INITIALIZE_FROM_SOURCE
SPECIFIED

Mpeg2QualityTuningLevel (enum)

Use Quality tuning level (Mpeg2QualityTuningLevel) to specify whether to use single-pass or multipass video encoding.

SINGLE_PASS
MULTI_PASS

Mpeg2RateControlMode (enum)

Use Rate control mode (Mpeg2RateControlMode) to specify whether the bitrate is variable (vbr) or constant (cbr).

VBR
CBR
Mpeg2SceneChangeDetect (enum)

Scene change detection (inserts I-frames on scene changes).

DISABLED
ENABLED

Mpeg2Settings

minIInterval

Enforces separation between repeated (cadence) I-frames and I-frames inserted by Scene Change Detection. If a scene change I-frame is within I-interval frames of a cadence I-frame, the GOP is shrunk and/or stretched to the scene change I-frame. GOP stretch requires enabling lookahead as well as setting I-interval. The normal cadence resumes for the next GOP. This setting is only used when Scene Change Detect is enabled. Note: Maximum GOP stretch = GOP size + Min-I-interval - 1

Type: integer
Required: False
Minimum: 0
Maximum: 30

parNumerator

Pixel Aspect Ratio numerator.

Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

gopSizeUnits

Type: string
Required: False

hrdBufferSize

Size of buffer (HRD buffer model) in bits. For example, enter five megabits as 5000000.

Type: integer
Required: False
Minimum: 0
Maximum: 47185920

qualityTuningLevel

Type: string
Required: False

maxBitrate

Maximum bitrate in bits/second. For example, enter five megabits per second as 5000000.
Properties

Type: integer
Required: False
Minimum: 1000
Maximum: 300000000

bitrate

Average bitrate in bits/second. Required for VBR and CBR. For MS Smooth outputs, bitrates must be unique when rounded down to the nearest multiple of 1000.

Type: integer
Required: False
Minimum: 1000
Maximum: 288000000

spatialAdaptiveQuantization

Type: string
Required: False

slowPal

Type: string
Required: False

intraDcPrecision

Type: string
Required: False

codecProfile

Type: string
Required: False

softness

Softness. Selects quantizer matrix, larger values reduce high-frequency content in the encoded image.

Type: integer
Required: False
Minimum: 0
Maximum: 128

framerateControl

Type: string
Required: False

telecine

Type: string
Required: False

codecLevel
Type: string
Required: False

framerateConversionAlgorithm
Type: string
Required: False

temporalAdaptiveQuantization
Type: string
Required: False

hrdBufferInitialFillPercentage
Percentage of the buffer that should initially be filled (HRD buffer model).
Type: integer
Required: False
Minimum: 0
Maximum: 100

framerateNumerator
Framerate numerator - framerate is a fraction, e.g. 24000 / 1001 = 23.976 fps.
Type: integer
Required: False
Minimum: 24
Maximum: 60000

gopClosedCadence
Frequency of closed GOPs. In streaming applications, it is recommended that this be set to 1 so a decoder joining mid-stream will receive an IDR frame as quickly as possible. Setting this value to 0 will break output segmenting.
Type: integer
Required: False
Minimum: 0
Maximum: 2147483647

numberBFramesBetweenReferenceFrames
Number of B-frames between reference frames.
Type: integer
Required: False
Minimum: 0
Maximum: 7

framerateDenominator

Framerate denominator.

Type: integer
Required: False
Minimum: 1
Maximum: 1001

adaptiveQuantization

Type: string
Required: False

interlaceMode

Type: string
Required: False

gopSize

GOP Length (keyframe interval) in frames or seconds. Must be greater than zero.

Type: number
Required: False
Format: float
Minimum: 0.0

syntax

Type: string
Required: False

parDenominator

Pixel Aspect Ratio denominator.

Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

sceneChangeDetect

Type: string
Required: False

parControl

Type: string
**Properties**

**Required:** False

**rateControlMode**

**Type:** string  
**Required:** False

**Mpeg2SlowPal (enum)**

Enables Slow PAL rate conversion. 23.976fps and 24fps input is relabeled as 25fps, and audio is sped up correspondingly.

- DISABLED
- ENABLED

**Mpeg2SpatialAdaptiveQuantization (enum)**

Adjust quantization within each frame based on spatial variation of content complexity.

- DISABLED
- ENABLED

**Mpeg2Syntax (enum)**

Produces a Type D-10 compatible bitstream (SMPTE 356M-2001).

- DEFAULT
- D_10

**Mpeg2Telecine (enum)**

Only use Telecine (Mpeg2Telecine) when you set Framerate (Framerate) to 29.970. Set Telecine (Mpeg2Telecine) to Hard (hard) to produce a 29.97i output from a 23.976 input. Set it to Soft (soft) to produce 23.976 output and leave conversion to the player.

- NONE
- SOFT
- HARD

**Mpeg2TemporalAdaptiveQuantization (enum)**

Adjust quantization within each frame based on temporal variation of content complexity.

- DISABLED
- ENABLED

**MsSmoothAudioDeduplication (enum)**

COMBINE_DUPLICATE_STREAMS combines identical audio encoding settings across a Microsoft Smooth output group into a single audio stream.

- COMBINE_DUPLICATE_STREAMS
- NONE
MsSmoothEncryptionSettings

spekeKeyProvider

Type: SpekeKeyProvider (p. 340)
Required: True

MsSmoothGroupSettings

fragmentLength

Use Fragment length (FragmentLength) to specify the mp4 fragment sizes in seconds. Fragment length must be compatible with GOP size and framerate.

Type: integer
Required: True
Minimum: 1
Maximum: 2147483647

encryption

Type: MsSmoothEncryptionSettings (p. 330)
Required: False

audioDeduplication

Type: string
Required: False

manifestEncoding

Type: string
Required: False

destination

Use Destination (Destination) to specify the S3 output location and the output filename base. Destination accepts format identifiers. If you do not specify the base filename in the URI, the service will use the filename of the input file. If your job has multiple inputs, the service uses the filename of the first input file.

Type: string
Required: False
Pattern: ^s3:/\/

MsSmoothManifestEncoding (enum)

Use Manifest encoding (MsSmoothManifestEncoding) to specify the encoding format for the server and client manifest. Valid options are utf8 and utf16.

UTF8
UTF16
NielsenConfiguration

distributorId

Use Distributor ID (DistributorID) to specify the distributor ID that is assigned to your organization by Nielsen.

Type: string
Required: False

breakoutCode

Use Nielsen Configuration (NielsenConfiguration) to set the Nielsen measurement system breakout code. Supported values are 0, 3, 7, and 9.

Type: integer
Required: False
Minimum: 0
Maximum: 9

NoiseReducer

filter

Type: string
Required: True

filterSettings

Type: NoiseReducerFilterSettings (p. 332)
Required: False

spatialFilterSettings

Type: NoiseReducerSpatialFilterSettings (p. 332)
Required: False

NoiseReducerFilter (enum)

Use Noise reducer filter (NoiseReducerFilter) to select one of the following spatial image filtering functions. To use this setting, you must also enable Noise reducer (NoiseReducer). * Bilateral is an edge preserving noise reduction filter. * Mean (softest), Gaussian, Lanczos, and Sharpen (sharpest) are convolution filters. * Conserve is a min/max noise reduction filter. * Spatial is a frequency-domain filter based on JND principles.

BILATERAL
MEAN
GAUSSIAN
LANCZOS
SHARPEN
CONSERVE
SPATIAL
NoiseReducerFilterSettings

strength

Relative strength of noise reducing filter. Higher values produce stronger filtering.

Type: integer
Required: False
Minimum: 0
Maximum: 3

NoiseReducerSpatialFilterSettings

strength

Relative strength of noise reducing filter. Higher values produce stronger filtering.

Type: integer
Required: False
Minimum: 0
Maximum: 16

postFilterSharpenStrength

Specify strength of post noise reduction sharpening filter, with 0 disabling the filter and 3 enabling it at maximum strength.

Type: integer
Required: False
Minimum: 0
Maximum: 3

speed

The speed of the filter, from -2 (lower speed) to 3 (higher speed), with 0 being the nominal value.

Type: integer
Required: False
Minimum: -2
Maximum: 3

Output

extension

Use Extension (Extension) to specify the file extension for outputs in File output groups. If you do not specify a value, the service will use default extensions by container type as follows: * MPEG-2 transport stream, m2ts * Quicktime, mov * MXF container, mxf * MPEG-4 container, mp4 * No Container, the service will use codec extensions (e.g. AAC, H265, H265, AC3)

Type: string
Required: False
videoDescription

(VideoDescription) contains a group of video encoding settings. The specific video settings depend on the video codec you choose when you specify a value for Video codec (codec). Include one instance of (VideoDescription) per output.

Type: VideoDescription (p. 346)
Required: False

audioDescriptions

(AudioDescriptions) contains groups of audio encoding settings organized by audio codec. Include one instance of (AudioDescriptions) per output. (AudioDescriptions) can contain multiple groups of encoding settings.

Type: Array of type AudioDescription (p. 238)
Required: False

containerSettings

Type: ContainerSettings (p. 254)
Required: False

preset

Use Preset (Preset) to specify a preset for your transcoding settings. Provide the system or custom preset name. You can specify either Preset (Preset) or Container settings (ContainerSettings), but not both.

Type: string
Required: False

outputSettings

Type: OutputSettings (p. 335)
Required: False

captionDescriptions

(CaptionDescriptions) contains groups of captions settings. For each output that has captions, include one instance of (CaptionDescriptions). (CaptionDescriptions) can contain multiple groups of captions settings.

Type: Array of type CaptionDescription (p. 245)
Required: False

nameModifier

Use Name modifier (NameModifier) to have the service add a string to the end of each output filename. You specify the base filename as part of your destination URI. When you create multiple outputs in the same output group, Name modifier (NameModifier) is required. Name modifier also accepts format identifiers. For DASH ISO outputs, if you use the format identifiers $Number$ or $Time$ in one output, you must use them in the same way in all outputs of the output group.

Type: string
Required: False

OutputChannelMapping

inputChannels

List of input channels

Type: Array of type integer
Required: True

OutputGroup

outputs

This object holds groups of encoding settings, one group of settings per output.

Type: Array of type Output (p. 332)
Required: True

outputGroupSettings

Type: OutputGroupSettings (p. 334)
Required: True

name

Name of the output group

Type: string
Required: False

customName

Use Custom Group Name (CustomName) to specify a name for the output group. This value is displayed on the console and can make your job settings JSON more human-readable. It does not affect your outputs. Use up to twelve characters that are either letters, numbers, spaces, or underscores.

Type: string
Required: False

OutputGroupSettings

dashIsoGroupSettings

Type: DashIsoGroupSettings (p. 255)
Required: False

fileGroupSettings

Type: FileGroupSettings (p. 269)
**Properties**

**Required:** False

**msSmoothGroupSettings**

**Type:** MsSmoothGroupSettings (p. 330)
**Required:** False

**cmafGroupSettings**

**Type:** CmafGroupSettings (p. 249)
**Required:** False

**type**

**Type:** string
**Required:** True

**hlsGroupSettings**

**Type:** HlsGroupSettings (p. 294)
**Required:** False

**OutputGroupType (enum)**

Type of output group (File group, Apple HLS, DASH ISO, Microsoft Smooth Streaming, CMAF)

- HLS_GROUP_SETTINGS
- DASH_ISO_GROUP_SETTINGS
- FILE_GROUP_SETTINGS
- MS_SMOOTH_GROUP_SETTINGS
- CMAF_GROUP_SETTINGS

**OutputSdt (enum)**

Selects method of inserting SDT information into output stream. "Follow input SDT" copies SDT information from input stream to output stream. "Follow input SDT if present" copies SDT information from input stream to output stream if SDT information is present in the input, otherwise it will fall back on the user-defined values. Enter "SDT Manually" means user will enter the SDT information. "No SDT" means output stream will not contain SDT information.

- SDT_FOLLOW
- SDT_FOLLOW_IF_PRESENT
- SDT_MANUAL
- SDT_NONE

**OutputSettings**

**hlsSettings**

**Type:** HlsSettings (p. 298)
**Required:** False
**ProresCodecProfile** (enum)

Use Profile (ProResCodecProfile) to specify the type of Apple ProRes codec to use for this output.

- `APPLE_PRORES_422`
- `APPLE_PRORES_422_HQ`
- `APPLE_PRORES_422_LT`
- `APPLE_PRORES_422_PROXY`

**ProresFramerateControl** (enum)

If you are using the console, use the Framerate setting to specify the framerate for this output. If you want to keep the same framerate as the input video, choose Follow source. If you want to do framerate conversion, choose a framerate from the dropdown list or choose Custom. The framerates shown in the dropdown list are decimal approximations of fractions. If you choose Custom, specify your framerate as a fraction. If you are creating your transcoding job specification as a JSON file without the console, use FramerateControl to specify which value the service uses for the framerate for this output. Choose `INITIALIZE_FROM_SOURCE` if you want the service to use the framerate from the input. Choose `SPECIFIED` if you want the service to use the framerate you specify in the settings FramerateNumerator and FramerateDenominator.

- `INITIALIZE_FROM_SOURCE`
- `SPECIFIED`

**ProresFramerateConversionAlgorithm** (enum)

When set to INTERPOLATE, produces smoother motion during framerate conversion.

- `DUPLICATE_DROP`
- `INTERPOLATE`

**ProresInterlaceMode** (enum)

Use Interlace mode (InterlaceMode) to choose the scan line type for the output. * Top Field First (TOP_FIELD) and Bottom Field First (BOTTOM_FIELD) produce interlaced output with the entire output having the same field polarity (top or bottom first). * Follow, Default Top (FOLLOW_TOP_FIELD) and Follow, Default Bottom (FOLLOW_BOTTOM_FIELD) use the same field polarity as the source. Therefore, behavior depends on the input scan type. - If the source is interlaced, the output will be interlaced with the same polarity as the source (it will follow the source). The output could therefore be a mix of "top field first" and "bottom field first". - If the source is progressive, the output will be interlaced with "top field first" or "bottom field first" polarity, depending on which of the Follow options you chose.

- `PROGRESSIVE`
- `TOP_FIELD`
- `BOTTOM_FIELD`
- `FOLLOW_TOP_FIELD`
- `FOLLOW_BOTTOM_FIELD`

**ProresParControl** (enum)

Use (ProresParControl) to specify how the service determines the pixel aspect ratio. Set to Follow source (INITIALIZE_FROM_SOURCE) to use the pixel aspect ratio from the input. To specify a different pixel aspect ratio: Using the console, choose it from the dropdown menu. Using the API, set ProresParControl to (SPECIFIED) and provide for (ParNumerator) and (ParDenominator).
INITIALIZE_FROM_SOURCE
SPECIFIED

**ProresSettings**

**framerateDenominator**
Framerate denominator.

*Type: integer*
*Required: False*
*Minimum: 1*
*Maximum: 2147483647*

**slowPal**

*Type: string*
*Required: False*

**framerateControl**

*Type: string*
*Required: False*

**telecine**

*Type: string*
*Required: False*

**framerateConversionAlgorithm**

*Type: string*
*Required: False*

**interlaceMode**

*Type: string*
*Required: False*

**parNumerator**
Pixel Aspect Ratio numerator.

*Type: integer*
*Required: False*
*Minimum: 1*
*Maximum: 2147483647*

**codecProfile**

*Type: string*
Required: False

parDenominator
Pixel Aspect Ratio denominator.

Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

parControl

Type: string
Required: False

framerateNumerator
When you use the API for transcode jobs that use framerate conversion, specify the framerate as a fraction. For example, 24000 / 1001 = 23.976 fps. Use FramerateNumerator to specify the numerator of this fraction. In this example, use 24000 for the value of FramerateNumerator.

Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

ProresSlowPal (enum)
Enables Slow PAL rate conversion. 23.976fps and 24fps input is relabeled as 25fps, and audio is sped up correspondingly.

DISABLED
ENABLED

ProresTelecine (enum)
Only use Telecine (ProresTelecine) when you set Framerate (Framerate) to 29.970. Set Telecine (ProresTelecine) to Hard (hard) to produce a 29.97i output from a 23.976 input. Set it to Soft (soft) to produce 23.976 output and leave conversion to the player.

NONE
HARD

Rectangle

width
Width of rectangle in pixels.

Type: integer
Required: True
Minimum: -2147483648

338
Properties

**Maximum**: 2,147,483,647

**x**

The distance, in pixels, between the rectangle and the left edge of the video frame.

- **Type**: integer
- **Required**: True
- **Minimum**: -2,147,483,648
- **Maximum**: 2,147,483,647

**y**

The distance, in pixels, between the rectangle and the top edge of the video frame.

- **Type**: integer
- **Required**: True
- **Minimum**: -2,147,483,648
- **Maximum**: 2,147,483,647

**height**

Height of rectangle in pixels.

- **Type**: integer
- **Required**: True
- **Minimum**: -2,147,483,648
- **Maximum**: 2,147,483,647

**RemixSettings**

**channelsOut**

Specify the number of channels in this output after remixing. Valid values: 1, 2, 4, 6, 8

- **Type**: integer
- **Required**: True
- **Minimum**: 1
- **Maximum**: 8

**channelMapping**

- **Type**: ChannelMapping (p. 248)
- **Required**: True

**channelsIn**

Specify the number of audio channels from your input that you want to use in your output. With remixing, you might combine or split the data in these channels, so the number of channels in your final output might be different.

- **Type**: integer
- **Required**: True
**Properties**

**Minimum**: 1  
**Maximum**: 16

**RespondToAfd (enum)**

Use Respond to AFD (RespondToAfd) to specify how the service changes the video itself in response to AFD values in the input. * Choose Respond to clip the input video frame according to the AFD value, input display aspect ratio, and output display aspect ratio. * Choose Passthrough to include the input AFD values. Do not choose this when AfdSignaling is set to (NONE). A preferred implementation of this workflow is to set RespondToAfd to (NONE) and set AfdSignaling to (AUTO). * Choose None to remove all input AFD values from this output.

- NONE
- RESPOND
- PASSTHROUGH

**ScalingBehavior (enum)**

Applies only if your input aspect ratio is different from your output aspect ratio. Enable Stretch to output (StretchToOutput) to have the service stretch your video image to fit. Leave this setting disabled to allow the service to letterbox your video instead. This setting overrides any positioning value you specify elsewhere in the job.

- DEFAULT
- STRETCH_TO_OUTPUT

**SccDestinationFramerate (enum)**

Set Framerate (SccDestinationFramerate) to make sure that the captions and the video are synchronized in the output. Specify a framerate that matches the framerate of the associated video. If the video framerate is 29.97, choose 29.97 dropframe (FRAMERATE_29_97_DROPFRAME) only if the video has video_insertion=true and drop_frame_timecode=true; otherwise, choose 29.97 non-dropframe (FRAMERATE_29_97_NON_DROPFRAME).

- FRAMERATE_23_97
- FRAMERATE_24
- FRAMERATE_29_97_DROPFRAME
- FRAMERATE_29_97_NON_DROPFRAME

**SccDestinationSettings**

**framerate**

- **Type**: string  
- **Required**: False

**SpekeKeyProvider**

**resourceld**

The SPEKE-compliant server uses Resource ID (Resourceld) to identify content.

- **Type**: string
Properties

Required: True
Pattern: ^\[\w-]+$  

**systemIds**

Relates to SPEKE implementation. DRM system identifiers. DASH output groups support a max of two system ids. Other group types support one system id.

Type: Array of type string
Required: True

**url**

Use URL (Url) to specify the SPEKE-compliant server that will provide keys for content.

Type: string
Required: True
Format: uri
Pattern: ^https:/\/

**StaticKeyProvider**

**keyFormat Versions**

Relates to DRM implementation. Either a single positive integer version value or a slash delimited list of version values (1/2/3).

Type: string
Required: False
Pattern: ^\d+(\/\d+)*$

**keyFormat**

Relates to DRM implementation. Sets the value of the KEYFORMAT attribute. Must be 'identity' or a reverse DNS string. May be omitted to indicate an implicit value of 'identity'.

Type: string
Required: False
Pattern: ^(identity|[A-Za-z]{2,6}(\.[A-Za-z0-9-]{1,63}+)\d+)$

**staticKeyValue**

Relates to DRM implementation. Use a 32-character hexadecimal string to specify Key Value (StaticKeyValue).

Type: string
Required: True
Pattern: ^[A-Za-z0-9]{32}$

**url**

Relates to DRM implementation. The location of the license server used for protecting content.

Type: string
Required: True
Format: uri

TeletextDestinationSettings

pageNumber

Set pageNumber to the Teletext page number for the destination captions for this output. This value must be a three-digit hexadecimal string; strings ending in -FF are invalid. If you are passing through the entire set of Teletext data, do not use this field.

Type: string
Required: False
Pattern: ^[1-8][0-9a-fA-F][0-9a-eA-E]$
TimecodeBurninPosition (enum)

Use Position (Position) under under Timecode burn-in (TimecodeBurnIn) to specify the location the
burned-in timecode on output video.

- TOP_CENTER
- TOP_LEFT
- TOP_RIGHT
- MIDDLE_LEFT
- MIDDLE_CENTER
- MIDDLE_RIGHT
- BOTTOM_LEFT
- BOTTOM_CENTER
- BOTTOM_RIGHT

TimecodeConfig

timestampOffset

Only applies to outputs that support program-date-time stamp. Use Timestamp offset
(TimestampOffset) to overwrite the timecode date without affecting the time and frame number.
Provide the new date as a string in the format "yyyy-mm-dd". To use Time stamp offset, you must also
enable Insert program-date-time (InsertProgramDateTime) in the output settings. For example, if the
date part of your timecodes is 2002-1-25 and you want to change it to one year later, set Timestamp
offset (TimestampOffset) to 2003-1-25.

  Type: string
  Required: False
  Pattern: ^([0-9]{4})-(0[1-9]|1[0-2])-(0[1-9]|12)[0-9][0-9]3[01]\$ 

anchor

If you use an editing platform that relies on an anchor timecode, use Anchor Timecode (Anchor) to
specify a timecode that will match the input video frame to the output video frame. Use 24-hour
format with frame number, (HH:MM:SS:FF) or (HH:MM:SS;FF). This setting ignores framerate conversion.
System behavior for Anchor Timecode varies depending on your setting for Source (TimecodeSource).
* If Source (TimecodeSource) is set to Specified Start (SPECIFIEDSTART), the first input frame is the
specified value in Start Timecode (Start). Anchor Timecode (Anchor) and Start Timecode (Start) are used
calculate output timecode. * If Source (TimecodeSource) is set to Start at 0 (ZEROBASED) the first frame
is 00:00:00:00. * If Source (TimecodeSource) is set to Embedded (EMBEDDED), the first frame is the
timecode value on the first input frame of the input.

  Type: string
  Required: False
  Format: timecode
  Pattern: ^([01][0-9]|2[0-4]):[0-5][0-9]:[0-5][0-9][0-9];[0-9][0-9][0-9][0-9]\$ 

start

Only use when you set Source (TimecodeSource) to Specified start (SPECIFIEDSTART). Use Start
timecode (Start) to specify the timecode for the initial frame. Use 24-hour format with frame number,
(HH:MM:SS:FF) or (HH:MM:SS;FF).

  Type: string
  Required: False
  Format: timecode
Pattern: ^((\[01][0-9]|2[0-4]):[0-5][0-9]):[0-5][0-9]:[;][0-9]{2}$

source

Type: string  
Required: False

TimecodeSource (enum)

Use Source (TimecodeSource) to set how timecodes are handled within this job. To make sure that your video, audio, captions, and markers are synchronized and that time-based features, such as image inserter, work correctly, choose the Timecode source option that matches your assets. All timecodes are in a 24-hour format with frame number (HH:MM:SS:FF). * Embedded (EMBEDDED) - Use the timecode that is in the input video. If no embedded timecode is in the source, the service will use Start at 0 (ZEROBASED) instead. * Start at 0 (ZEROBASED) - Set the timecode of the initial frame to 00:00:00:00. * Specified Start (SPECIFIEDSTART) - Set the timecode of the initial frame to a value other than zero. You use Start timecode (Start) to provide this value.

EMBEDDED  
ZEROBASED  
SPECIFIEDSTART

TimedMetadata (enum)

Applies only to HLS outputs. Use this setting to specify whether the service inserts the ID3 timed metadata from the input in this output.

PASSTHROUGH  
NONE

TimedMetadataInsertion

id3Insertions

Id3Insertions contains the array of Id3Insertion instances.

Type: Array of type Id3Insertion (p. 299)  
Required: True

TtmlDestinationSettings

stylePassthrough

Type: string  
Required: False

TtmlStylePassthrough (enum)

Pass through style and position information from a TTML-like input source (TTML, SMPTE-TT, CFF-TT) to the CFF-TT output or TTML output.

ENABLED  
DISABLED
**Type (enum)**

- SYSTEM
- CUSTOM

**UpdateJobTemplateRequest**

**settings**

*Type:* JobTemplateSettings (p. 306)

*Required:* False

**name**

The name of the job template you are modifying

*Type:* string

*Required:* False

**description**

The new description for the job template, if you are changing it.

*Type:* string

*Required:* False

**category**

The new category for the job template, if you are changing it.

*Type:* string

*Required:* False

**queue**

The new queue for the job template, if you are changing it.

*Type:* string

*Required:* False

**UpdateJobTemplateResponse**

**jobTemplate**

*Type:* JobTemplate (p. 304)

*Required:* False

**VideoCodec (enum)**

Type of video codec

- FRAME_CAPTURE
- H_264
VideoCodecSettings

h265Settings
  Type: H265Settings (p. 283)
  Required: False

codec
  Type: string
  Required: True

proresSettings
  Type: ProresSettings (p. 337)
  Required: False

mpeg2Settings
  Type: Mpeg2Settings (p. 325)
  Required: False

frameCaptureSettings
  Type: FrameCaptureSettings (p. 270)
  Required: False

h264Settings
  Type: H264Settings (p. 274)
  Required: False

VideoDescription

fixedAfd
  Applies only if you set AFD Signaling(AfdSignaling) to Fixed (FIXED). Use Fixed (FixedAfd) to specify a four-bit AFD value which the service will write on all frames of this video output.

  Type: integer
  Required: False
  Minimum: 0
  Maximum: 15

scalingBehavior
  Type: string
  Required: False
respondToAfd
   Type: string
   Required: False

codecSettings
   Type: VideoCodecSettings (p. 346)
   Required: True

afdSignaling
   Type: string
   Required: False

colorMetadata
   Type: string
   Required: False

timecodeInsertion
   Type: string
   Required: False

width
Use Width (Width) to define the video resolution width, in pixels, for this output. If you don't provide a value here, the service will use the input width.

   Type: integer
   Required: False
   Minimum: 32
   Maximum: 4096

videoPreprocessors
Find additional transcoding features under Preprocessors (VideoPreprocessors). Enable the features at each output individually. These features are disabled by default.

   Type: VideoPreprocessor (p. 348)
   Required: False

antiAlias
   Type: string
   Required: False

position
Use Position (Position) to point to a rectangle object to define your position. This setting overrides any other aspect ratio.
Type: Rectangle (p. 338)
Required: False

sharpness

Use Sharpness (Sharpness) setting to specify the strength of anti-aliasing. This setting changes the width of the anti-alias filter kernel used for scaling. Sharpness only applies if your output resolution is different from your input resolution, and if you set Anti-alias (AntiAlias) to ENABLED. 0 is the softest setting, 100 the sharpest, and 50 recommended for most content.

Type: integer
Required: False
Minimum: 0
Maximum: 100

crop

Applies only if your input aspect ratio is different from your output aspect ratio. Use Input cropping rectangle (Crop) to specify the video area the service will include in the output. This will crop the input source, causing video pixels to be removed on encode. Do not use this setting if you have enabled Stretch to output (stretchToOutput) in your output settings.

Type: Rectangle (p. 338)
Required: False

dropFrameTimecode

Type: string
Required: False

height

Use the Height (Height) setting to define the video resolution height for this output. Specify in pixels. If you don't provide a value here, the service will use the input height.

Type: integer
Required: False
Minimum: 32
Maximum: 2160

VideoPreprocessor

noiseReducer

Enable the Noise reducer (NoiseReducer) feature to remove noise from your video output if necessary. Enable or disable this feature for each output individually. This setting is disabled by default.

Type: NoiseReducer (p. 331)
Required: False

timecodeBurnin

Timecode burn-in (TimecodeBurnIn) -- Burns the output timecode and specified prefix into the output.
Properties

Type: TimecodeBurnin (p. 342)
Required: False

colorCorrector
Enable the Color corrector (ColorCorrector) feature if necessary. Enable or disable this feature for each output individually. This setting is disabled by default.

Type: ColorCorrector (p. 252)
Required: False

deinterlacer
Use Deinterlacer (Deinterlacer) to produce smoother motion and a clearer picture.

Type: Deinterlacer (p. 257)
Required: False

imageInserter
Enable the Image inserter (ImageInserter) feature to include a graphic overlay on your video. Enable or disable this feature for each output individually. This setting is disabled by default.

Type: ImageInserter (p. 299)
Required: False

VideoSelector

colorSpace

Type: string
Required: False

hdr10Metadata

Type: Hdr10Metadata (p. 289)
Required: False

programNumber
Selects a specific program from within a multi-program transport stream. Note that Quad 4K is not currently supported.

Type: integer
Required: False
Minimum: -2147483648
Maximum: 2147483647

pid
Use PID (Pid) to select specific video data from an input file. Specify this value as an integer; the system automatically converts it to the hexadecimal value. For example, 257 selects PID 0x101. A PID, or packet identifier, is an identifier for a set of data in an MPEG-2 transport stream container.
**Properties**

**Type**: integer  
**Required**: False  
**Minimum**: 1  
**Maximum**: 2147483647

**colorSpaceUsage**

**Type**: string  
**Required**: False

**VideoTimecodeInsertion (enum)**

Applies only to H.264, H.265, MPEG2, and ProRes outputs. Only enable Timecode insertion when the input framerate is identical to the output framerate. To include timecodes in this output, set Timecode insertion (VideoTimecodeInsertion) to PIC_TIMING_SEI. To leave them out, set it to DISABLED. Default is DISABLED. When the service inserts timecodes in an output, by default, it uses any embedded timecodes from the input. If none are present, the service will set the timecode for the first output frame to zero. To change this default behavior, adjust the settings under Timecode configuration (TimecodeConfig). In the console, these settings are located under Job > Job settings > Timecode configuration. Note - Timecode source under input settings (InputTimecodeSource) does not affect the timecodes that are inserted in the output. Source under Job settings > Timecode configuration (TimecodeSource) does.

- DISABLED
- PIC_TIMING_SEI

**WavFormat (enum)**

The service defaults to using RIFF for WAV outputs. If your output audio is likely to exceed 4 GB in file size, or if you otherwise need the extended support of the RF64 format, set your output WAV file format to RF64.

- RIFF
- RF64

**WavSettings**

**channels**

Set Channels to specify the number of channels in this output audio track. With WAV, valid values 1, 2, 4, and 8. In the console, these values are Mono, Stereo, 4-Channel, and 8-Channel, respectively.

**Type**: integer  
**Required**: False  
**Minimum**: 1  
**Maximum**: 8

**bitDepth**

Specify Bit depth (BitDepth), in bits per sample, to choose the encoding quality for this audio track.

**Type**: integer  
**Required**: False  
**Minimum**: 16
Maximum: 24

format

Type: string
Required: False

sampleRate

Sample rate in Hz.

Type: integer
Required: False
Minimum: 8000
Maximum: 192000

Jobs

URI

/2017-08-29/jobs

HTTP Methods

GET

Operation ID: ListJobs

Retrieve a JSON array of up to twenty of your most recently created jobs. This array includes in-process, completed, and errored jobs. This will return the jobs themselves, not just a list of the jobs. To retrieve the twenty next most recent jobs, use the nextToken string returned with the array.

Query Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>String</td>
<td>False</td>
<td></td>
</tr>
<tr>
<td>nextToken</td>
<td>String</td>
<td>False</td>
<td></td>
</tr>
<tr>
<td>maxResults</td>
<td>String</td>
<td>False</td>
<td></td>
</tr>
<tr>
<td>order</td>
<td>String</td>
<td>False</td>
<td></td>
</tr>
<tr>
<td>queue</td>
<td>String</td>
<td>False</td>
<td></td>
</tr>
</tbody>
</table>

Responses

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>ListJobsResponse (p. 363)</td>
<td>200 response</td>
</tr>
<tr>
<td>400</td>
<td>ExceptionBody (p. 384)</td>
<td>BadRequestException</td>
</tr>
</tbody>
</table>
### Status Codes

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>ExceptionBody (p. 384)</td>
<td>InternalServiceException</td>
</tr>
<tr>
<td>403</td>
<td>ExceptionBody (p. 384)</td>
<td>AccessDeniedException</td>
</tr>
<tr>
<td>404</td>
<td>ExceptionBody (p. 384)</td>
<td>ResourceNotFoundException</td>
</tr>
<tr>
<td>429</td>
<td>ExceptionBody (p. 384)</td>
<td>LimitExceededException</td>
</tr>
<tr>
<td>409</td>
<td>ExceptionBody (p. 384)</td>
<td>ResourceInUseException</td>
</tr>
</tbody>
</table>

### POST

**Operation ID: CreateJob**

Create a new transcoding job. For information about jobs and job settings, see the User Guide at http://docs.aws.amazon.com/mediaconvert/latest/ug/what-is.html

### Responses

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>CreateJobResponse (p. 373)</td>
<td>201 response</td>
</tr>
<tr>
<td>400</td>
<td>ExceptionBody (p. 384)</td>
<td>BadRequestException</td>
</tr>
<tr>
<td>500</td>
<td>ExceptionBody (p. 384)</td>
<td>InternalServiceException</td>
</tr>
<tr>
<td>403</td>
<td>ExceptionBody (p. 384)</td>
<td>AccessDeniedException</td>
</tr>
<tr>
<td>404</td>
<td>ExceptionBody (p. 384)</td>
<td>ResourceNotFoundException</td>
</tr>
<tr>
<td>429</td>
<td>ExceptionBody (p. 384)</td>
<td>LimitExceededException</td>
</tr>
<tr>
<td>409</td>
<td>ExceptionBody (p. 384)</td>
<td>ResourceInUseException</td>
</tr>
</tbody>
</table>

### Schemas

#### Request Bodies

##### Example GET

```json
{
    "nextToken (p. 465)": "string",
    "maxResults (p. 465)": integer,
    "queue (p. 465)": "string",
    "status (p. 465)": enum,
    "order (p. 465)": enum
}
```

##### Example POST

```json
{
    "settings (p. 408)": {
        "timecodeConfig (p. 460)": {
```
"timestampOffset (p. 499)": "string",
"anchor (p. 499)": "string",
"start (p. 499)": "string",
"source (p. 500)": enum
},
"adAvailOffset (p. 460)": integer,
"outputGroups (p. 460)": [
{
"outputs (p. 490)": [
{
"extension (p. 488)": "string",
"videoDescription (p. 488)": {
"fixedAfd (p. 502)": integer,
"scalingBehavior (p. 502)": enum,
"respondToAfd (p. 502)": enum,
"codecSettings (p. 502)": {
"h265Settings (p. 501)": {
"slices (p. 436)": integer,
"minInterval (p. 437)": integer,
"parNumerator (p. 437)": integer,
"flickerAdaptiveQuantization (p. 437)": enum,
"gopSizeUnits (p. 437)": enum,
"hrdBufferSize (p. 437)": integer,
"qualityTuningLevel (p. 437)": enum,
"maxBitrate (p. 437)": integer,
"bitrate (p. 437)": integer,
"spatialAdaptiveQuantization (p. 438)": enum,
"temporalIds (p. 438)": enum,
"slowPal (p. 438)": enum,
"tiles (p. 438)": enum,
"writeM4PackagingType (p. 438)": enum,
"codecProfile (p. 438)": enum,
"alternateTransferFunctionSei (p. 438)": enum,
"unregisteredSeiTimecode (p. 439)": enum,
"framerateControl (p. 439)": enum,
"telecine (p. 439)": enum,
"framerateConversionAlgorithm (p. 439)": enum,
"numberReferenceFrames (p. 439)": integer,
"temporalAdaptiveQuantization (p. 439)": enum,
"hrdBufferInitialFillPercentage (p. 439)": integer,
"framerateDenominator (p. 439)": enum,
"framerateConversionAlgorithm (p. 439)": enum,
"parNumerator (p. 440)": integer,
"parDenominator (p. 440)": integer,
"sceneChangeDetect (p. 441)": enum,
"parControl (p. 441)": enum,
"rateControlMode (p. 441)": enum
},
"codec (p. 501)": enum,
"proresSettings (p. 501)": {
"framerateDenominator (p. 493)": integer,
"slowPal (p. 493)": enum,
"framerateControl (p. 493)": enum,
"telecine (p. 493)": enum,
"framerateConversionAlgorithm (p. 493)": enum,
"parNumerator (p. 493)": integer,
"codecProfile (p. 493)": enum,
"parDenominator (p. 494)": integer,
"parControl (p. 494)" : enum,
"framerateNumerator (p. 494)" : integer
},
"mpeg2Settings (p. 502)" : {
"minIInterval (p. 480)" : integer,
"parNumerator (p. 480)" : integer,
"gopSizeUnits (p. 481)" : enum,
"hrdBufferSize (p. 481)" : integer,
"qualityTuningLevel (p. 481)" : enum,
"maxBitrate (p. 481)" : integer,
"bitrate (p. 481)" : integer,
"spatialAdaptiveQuantization (p. 481)" : enum,
"slowPal (p. 481)" : enum,
"intraDcPrecision (p. 482)" : enum,
"codecProfile (p. 482)" : enum,
"softness (p. 482)" : integer,
"framerateControl (p. 482)" : enum,
"telecine (p. 482)" : enum,
"codecLevel (p. 482)" : enum,
"framerateConversionAlgorithm (p. 482)" : enum,
"temporalAdaptiveQuantization (p. 482)" : enum,
"hrdBufferInitialFillPercentage (p. 482)" : integer,
"framerateNumerator (p. 483)" : integer,
"gopClosedCadence (p. 483)" : integer,
"numberBFramesBetweenReferenceFrames (p. 483)" : integer,
"framerateDenominator (p. 483)" : integer,
"adaptiveQuantization (p. 483)" : enum,
"interlaceMode (p. 483)" : enum,
"gopSize (p. 483)" : number,
"syntax (p. 484)" : enum,
"parDenominator (p. 484)" : integer,
"sceneChangeDetect (p. 484)" : enum,
"parControl (p. 484)" : enum,
"rateControlMode (p. 484)" : enum
},
"frameCaptureSettings (p. 502)" : {
"framerateDenominator (p. 424)" : integer,
"maxCaptures (p. 424)" : integer,
"framerateNumerator (p. 424)" : integer,
"quality (p. 424)" : integer
},
"h264Settings (p. 502)" : {
"slices (p. 428)" : integer,
"minIInterval (p. 428)" : integer,
"parNumerator (p. 428)" : integer,
"flickerAdaptiveQuantization (p. 428)" : enum,
"gopSizeUnits (p. 428)" : enum,
"hrdBufferSize (p. 428)" : integer,
"qualityTuningLevel (p. 429)" : enum,
"maxBitrate (p. 429)" : integer,
"bitrate (p. 429)" : integer,
"spatialAdaptiveQuantization (p. 429)" : enum,
"slowPal (p. 429)" : enum,
"codecProfile (p. 429)" : enum,
"unregisteredSeiTimecode (p. 429)" : enum,
"softness (p. 429)" : integer,
"framerateControl (p. 430)" : enum,
"telecine (p. 430)" : enum,
"codecLevel (p. 430)" : enum,
"framerateConversionAlgorithm (p. 430)" : enum,
"numberReferenceFrames (p. 430)" : integer,
"temporalAdaptiveQuantization (p. 430)" : enum,
"repeatPps (p. 430)" : enum,
"hrdBufferInitialFillPercentage (p. 430)" : integer,
"framerateNumerator (p. 431)" : integer,
"gopClosedCadence (p. 431)" : integer,
"numberBFramesBetweenReferenceFrames (p. 431)": integer,
"framerateDenominator (p. 431)": integer,
"entropyEncoding (p. 431)": enum,
"fieldEncoding (p. 431)": enum,
"adaptiveQuantization (p. 432)": enum,
"interlaceMode (p. 432)": enum,
"gopSize (p. 432)": number,
"gopBReference (p. 432)": enum,
"syntax (p. 432)": enum,
"parDenominator (p. 432)": integer,
"sceneChangeDetect (p. 432)": enum,
"parControl (p. 432)": enum,
"rateControlMode (p. 432)": enum
},
"afdSignaling (p. 502)": enum,
"colorMetadata (p. 502)": enum,
"timecodeInsertion (p. 503)": enum,
"width (p. 503)": integer,
"videoPreprocessors (p. 503)": {
"noiseReducer (p. 504)": {
"filter (p. 486)": enum,
"filterSettings (p. 486)": {
"strength (p. 487)": integer
},
"spatialFilterSettings (p. 487)": {
"strength (p. 487)": integer,
"postFilterSharpenStrength (p. 487)": integer,
"speed (p. 488)": integer
}
},
"timecodeBurnin (p. 504)": {
"prefix (p. 498)": "string",
"fontSize (p. 498)": integer,
"position (p. 498)": enum
},
"colorCorrector (p. 505)": {
"saturation (p. 408)": integer,
"brightness (p. 409)": integer,
"hdr10Metadata (p. 405)": {
"redPrimaryY (p. 442)": integer,
"greenPrimaryY (p. 443)": integer,
"whitePointX (p. 443)": integer,
"maxLuminance (p. 443)": integer,
"greenPrimaryX (p. 443)": integer,
"whitePointY (p. 443)": integer,
"redPrimaryX (p. 444)": integer,
"bluePrimaryX (p. 444)": integer,
"bluePrimaryY (p. 444)": integer,
"maxFrameAverageLightLevel (p. 444)": integer,
"maxContentLightLevel (p. 444)": integer,
"minLuminance (p. 445)": integer
},
"contrast (p. 405)": integer,
"hue (p. 405)": integer,
"colorSpaceConversion (p. 406)": enum
},
"deinterlacer (p. 505)": {
"mode (p. 411)": enum,
"control (p. 411)": enum,
"algorithm (p. 411)": enum
},
"imageInserter (p. 505)": {
"insertableImages (p. 453)": [
{
"duration (p. 456)": integer,
"fadeOut (p. 456)" : integer,
"imageY (p. 457)" : integer,
"fadeIn (p. 457)" : integer,
"imageX (p. 457)" : integer,
"width (p. 457)" : integer,
"startTime (p. 457)" : "string",
"opacity (p. 457)" : integer,
"layer (p. 458)" : integer,
"height (p. 458)" : integer,
"imageInserterInput (p. 458)" : "string"
}
]
}
"antiAlias (p. 503)" : enum,
"position (p. 503)" : {
  "width (p. 494)" : integer,
  "x (p. 495)" : integer,
  "y (p. 495)" : integer,
  "height (p. 495)" : integer
},
"sharpness (p. 503)" : integer,
"crop (p. 503)" : {
  "width (p. 494)" : integer,
  "x (p. 495)" : integer,
  "y (p. 495)" : integer,
  "height (p. 495)" : integer
},
"dropFrameTimecode (p. 504)" : enum,
"height (p. 504)" : integer
},
"audioDescriptions (p. 488)" : [
  {
    "audioTypeControl (p. 391)" : enum,
    "languageCodeControl (p. 391)" : enum,
    "remixSettings (p. 391)" : {
      "channelsOut (p. 495)" : integer,
      "channelMapping (p. 495)" : {
        "outputChannels (p. 401)" : [
          {
            "inputChannels (p. 489)" : [integer
          ]
        ]
      }
    },
    "channelsIn (p. 495)" : integer
  },
  "audioType (p. 391)" : integer,
  "audioSourceName (p. 391)" : "string",
  "codecSettings (p. 392)" : {
    "codec (p. 390)" : enum,
    "wavSettings (p. 390)" : {
      "channels (p. 506)" : integer,
      "bitDepth (p. 506)" : integer,
      "format (p. 506)" : enum,
      "sampleRate (p. 507)" : integer
    },
    "aacSettings (p. 390)" : {
      "vbrQuality (p. 385)" : enum,
      "codecProfile (p. 385)" : enum,
      "codingMode (p. 385)" : enum,
      "specification (p. 386)" : enum,
      "bitrate (p. 386)" : integer,
      "rawFormat (p. 386)" : enum,
      "rateControlMode (p. 386)" : enum
    }
  }
]
"sampleRate (p. 386)" : integer,
"audioDescriptionBroadcasterMix (p. 386)" : enum,
"ac3Settings (p. 390)" : {
  "dynamicRangeCompressionProfile (p. 387)" : enum,
  "dialnorm (p. 388)" : integer,
  "codingMode (p. 388)" : enum,
  "metadataControl (p. 388)" : enum,
  "bitrate (p. 388)" : integer,
  "lfeFilter (p. 388)" : enum,
  "bitstreamMode (p. 388)" : enum,
  "sampleRate (p. 388)" : integer
},
"aiffSettings (p. 390)" : {
  "channels (p. 389)" : integer,
  "bitDepth (p. 389)" : integer,
  "sampleRate (p. 389)" : integer
},
"eac3Settings (p. 390)" : {
  "dialnorm (p. 419)" : integer,
  "passthroughControl (p. 419)" : enum,
  "metadataControl (p. 419)" : enum,
  "bitrate (p. 419)" : integer,
  "dynamicRangeCompressionRf (p. 419)" : enum,
  "sampleRate (p. 419)" : integer,
  "surroundExMode (p. 419)" : enum,
  "l1tRtSurroundMixLevel (p. 420)" : number,
  "dynamicRangeCompressionLine (p. 420)" : enum,
  "lfeControl (p. 420)" : enum,
  "codingMode (p. 420)" : enum,
  "surroundMode (p. 420)" : enum,
  "attenuationControl (p. 420)" : enum,
  "lfeFilter (p. 420)" : enum,
  "phaseControl (p. 420)" : enum,
  "l1tRtCenterMixLevel (p. 420)" : number,
  "dcFilter (p. 421)" : enum,
  "bitstreamMode (p. 421)" : enum,
  "surroundMode (p. 421)" : enum,
  "loRoSurroundMixLevel (p. 421)" : number,
  "loRoCenterMixLevel (p. 421)" : number
},
"mp2Settings (p. 390)" : {
  "channels (p. 477)" : integer,
  "bitrate (p. 477)" : integer,
  "sampleRate (p. 477)" : integer
},
"languageCode (p. 392)" : enum,
"streamName (p. 392)" : "string",
"audioNormalizationSettings (p. 392)" : {
  "targetLkfs (p. 393)" : number,
  "algorithmControl (p. 393)" : enum,
  "loudnessLogging (p. 393)" : enum,
  "peakCalculation (p. 393)" : enum,
  "correctionGateLevel (p. 394)" : integer,
  "algorithm (p. 394)" : enum
},
"customLanguageCode (p. 392)" : "string"
}]
"containerSettings (p. 488)" : {
  "container (p. 407)" : enum,
  "mp4Settings (p. 407)" : {
    "mp4MajorBrand (p. 478)" : "string",
    "moovPlacement (p. 478)" : enum,
    "csigAtom (p. 478)" : enum,
    "cslgAtom (p. 478)" : enum,
    "moovAtom (p. 478)" : enum,
    "esidAtom (p. 478)" : enum,
    "esidBlock (p. 478)" : enum,
    "mftEntry (p. 478)" : enum,
    "mftAtom (p. 478)" : enum,
    "mftEntry (p. 478)" : enum,
    "mftAtom (p. 478)" : enum
  }
}
"freeSpaceBox (p. 478)" : enum
,
"m3u8Settings (p. 407)" : {
  "pmtPid (p. 473)" : integer,
  "nielsenId3 (p. 473)" : enum,
  "pcrPid (p. 473)" : integer,
  "audioPids (p. 473)" : [ integer
    ],
  "audioFramesPerPes (p. 473)" : integer,
  "scte35Pid (p. 474)" : integer,
  "transportStreamId (p. 474)" : integer,
  "pcrControl (p. 474)" : enum,
  "videoPid (p. 474)" : integer,
  "privateMetadataPid (p. 474)" : integer,
  "pmtInterval (p. 474)" : integer,
  "patInterval (p. 475)" : integer,
  "programNumber (p. 475)" : integer,
  "timeMetadataPid (p. 475)" : integer,
  "timeMetadata (p. 475)" : enum,
  "scte35Source (p. 475)" : enum,
},
"m2tsSettings (p. 407)" : {
  "dvbTeletextPid (p. 468)" : integer,
  "nielsenId3 (p. 468)" : enum,
  "bitrate (p. 468)" : integer,
  "segmentationTime (p. 468)" : number,
  "audioPids (p. 468)" : [ integer
    ],
  "rateMode (p. 468)" : enum,
  "maxPcrInterval (p. 468)" : integer,
  "audioFramesPerPes (p. 469)" : integer,
  "ebpAudioInterval (p. 469)" : enum,
  "fragmentTime (p. 469)" : number,
  "scte35Pid (p. 469)" : integer,
  "pmtInterval (p. 469)" : integer,
  "privateMetadataPid (p. 469)" : integer,
  "segmentationStyle (p. 470)" : enum,
  "audioBufferModel (p. 470)" : enum,
  "programNumber (p. 470)" : integer,
  "dvbNitSettings (p. 470)" : {
    "networkName (p. 411)" : "string",
    "networkId (p. 412)" : integer,
    "nitInterval (p. 412)" : integer
  },
  "timeMetadataPid (p. 470)" : integer,
  "scte35Source (p. 470)" : enum,
  "bufferModel (p. 470)" : enum,
  "ebpPlacement (p. 471)" : enum,
  "dvbSdtSettings (p. 471)" : {
    "sdtInterval (p. 412)" : integer,
    "serviceName (p. 412)" : "string",
    "serviceProviderName (p. 412)" : "string",
    "outputSdt (p. 412)" : enum
  },
  "nullPacketBitrate (p. 471)" : number,
  "pcrPid (p. 471)" : integer,
  "minEbpInterval (p. 471)" : integer,
  "transportStreamId (p. 471)" : integer,
  "videoPid (p. 472)" : integer,
  "pcrControl (p. 472)" : enum,
  "esRateInPes (p. 472)" : enum,
  "segmentationMarkers (p. 472)" : enum,
  "dvbTdtSettings (p. 472)" : {
}
"tdtInterval (p. 417)": integer,
"patInterval (p. 472)": integer,
"dvbSubPids (p. 472)": [
    integer
],
"movSettings (p. 407)": {
    "reference (p. 476)": enum,
    "paddingControl (p. 476)": enum,
    "mpeg2FourCCControl (p. 476)": enum,
    "cslgAtom (p. 476)": enum,
    "clapAtom (p. 476)": enum
},
"f4vSettings (p. 407)": {
    "moovPlacement (p. 423)": enum
},
"preset (p. 489)": "string",
"outputSettings (p. 489)": {
    "hlsSettings (p. 491)": {
        "iFrameOnlyManifest (p. 451)": enum,
        "audioRenditionSets (p. 451)": "string",
        "audioTrackType (p. 452)": enum,
        "segmentModifier (p. 452)": "string",
        "audioGroupId (p. 452)": "string"
    }
},
"captionDescriptions (p. 489)": [
    {
        "captionSelectorName (p. 398)": "string",
        "languageDescription (p. 398)": "string",
        "destinationSettings (p. 399)": {
            "burninDestinationSettings (p. 399)": {
                "xPosition (p. 394)": integer,
                "backgroundColor (p. 395)": enum,
                "teletextSpacing (p. 395)": enum,
                "yPosition (p. 395)": integer,
                "backgroundOpacity (p. 395)": integer,
                "fontOpacity (p. 395)": integer,
                "fontResolution (p. 395)": integer,
                "shadowOpacity (p. 396)": integer,
                "shadowYOffset (p. 396)": integer,
                "outlineSize (p. 396)": integer,
                "outlineColor (p. 396)": enum,
                "fontSize (p. 396)": integer,
                "shadowXOffset (p. 396)": integer,
                "alignment (p. 397)": enum,
                "shadowColor (p. 397)": enum,
                "fontColor (p. 397)": enum
            },
            "teletextDestinationSettings (p. 399)": {
                "pageNumber (p. 498)": "string"
            },
            "ttmlDestinationSettings (p. 399)": {
                "stylePassthrough (p. 501)": enum
            }
        },
        "destinationType (p. 399)": enum,
        "dvbSubDestinationSettings (p. 399)": {
            "xPosition (p. 413)": integer,
            "backgroundColor (p. 413)": enum,
            "teletextSpacing (p. 413)": enum,
            "yPosition (p. 413)": integer,
            "backgroundOpacity (p. 413)": integer,
            "fontOpacity (p. 413)": integer,
            "fontResolution (p. 414)": integer,
"shadowOpacity (p. 414)": integer,
"shadowXOffset (p. 414)": integer,
"outlineSize (p. 414)": integer,
"outlineColor (p. 414)": enum,
"fontSize (p. 414)": integer,
"shadowXOffset (p. 415)": integer,
"alignment (p. 415)": enum,
"shadowColor (p. 415)": enum,
"foregroundColor (p. 415)": enum
},
"sccDestinationSettings (p. 399)": {
  "framerate (p. 496)": enum
},
"languageCode (p. 399)": enum,
"customLanguageCode (p. 399)": "string"
},
"nameModifier (p. 489)": "string"
}],
"outputGroupSettings (p. 490)": {
  "dashIsoGroupSettings (p. 490)": {
    "fragmentLength (p. 409)": integer,
    "baseUrl (p. 409)": "string",
    "minBufferTime (p. 409)": integer,
    "encryption (p. 409)": {
      "spekeKeyProvider (p. 409)": {
        "resourceId (p. 496)": "string",
        "systemIds (p. 497)": ["string"
      ],
      "url (p. 497)": "string"
    },
    "destination (p. 409)": "string",
    "segmentLength (p. 410)": integer,
    "segmentControl (p. 410)": enum,
    "hbbtvCompliance (p. 410)": enum
  },
  "fileGroupSettings (p. 490)": {
    "destination (p. 423)": "string"
  },
  "msSmoothGroupSettings (p. 491)": {
    "fragmentLength (p. 485)": integer,
    "encryption (p. 485)": {
      "spekeKeyProvider (p. 485)": {
        "resourceId (p. 496)": "string",
        "systemIds (p. 497)": ["string"
      ],
      "url (p. 497)": "string"
    },
    "audioDeduplication (p. 485)": enum,
    "manifestEncoding (p. 486)": enum,
    "destination (p. 486)": "string"
  },
  "cmafGroupSettings (p. 491)": {
    "writeHlsManifest (p. 402)": enum,
    "writeDashManifest (p. 402)": enum,
    "fragmentLength (p. 402)": integer,
    "streamInfResolution (p. 402)": enum,
    "minBufferTime (p. 402)": integer,
    "clientCache (p. 403)": enum,
    "codecSpecification (p. 403)": enum,
...
"destination (p. 403)": "string",
"segmentControl (p. 403)": enum,
"baseUrl (p. 403)": "string",
"encryption (p. 403)": {
  "initializationVectorInManifest (p. 401)": enum,
  "constantInitializationVector (p. 401)": "string",
  "staticKeyProvider (p. 401)": {
    "keyFormatVersions (p. 407)": "string",
    "keyFormat (p. 407)": "string",
    "staticKeyValue (p. 407)": "string",
    "url (p. 407)": "string"
  },
  "type (p. 402)": enum,
  "encryptionMethod (p. 402)": enum
},
"manifestCompression (p. 403)": enum,
"segmentLength (p. 403)": integer,
"manifestDurationFormat (p. 404)": enum
},
"type (p. 491)": enum,
"hlsGroupSettings (p. 491)": {
  "segmentsPerSubdirectory (p. 447)": integer,
  "streamInfResolution (p. 447)": enum,
  "timestampDeltaMilliseconds (p. 448)": integer,
  "outputSelection (p. 448)": enum,
  "captionLanguageMappings (p. 448)": [
    {
      "languageDescription (p. 445)": "string",
      "captionChannel (p. 445)": integer,
      "languageCode (p. 445)": enum,
      "customLanguageCode (p. 446)": "string"
    }
  ],
  "clientCache (p. 448)": enum,
  "codecSpecification (p. 448)": enum,
  "destination (p. 448)": "string",
  "timedMetadataId3Frame (p. 448)": enum,
  "segmentControl (p. 448)": enum,
  "timedMetadataId3Period (p. 449)": integer,
  "captionLanguageSetting (p. 449)": enum,
  "minSegmentLength (p. 449)": integer,
  "directoryStructure (p. 449)": enum,
  "programDateTime (p. 449)": enum,
  "baseUrl (p. 449)": "string",
  "encryption (p. 449)": {
    "initializationVectorInManifest (p. 446)": enum,
    "constantInitializationVector (p. 446)": "string",
    "staticKeyProvider (p. 446)": {
      "keyFormatVersions (p. 447)": "string",
      "keyFormat (p. 447)": "string",
      "staticKeyValue (p. 447)": "string",
      "url (p. 447)": "string"
    },
    "type (p. 447)": enum,
    "encryptionMethod (p. 447)": enum,
    "spekeKeyProvider (p. 447)": {
      "resourceId (p. 446)": "string",
      "systemIds (p. 447)": [
        "string"
      ],
      "url (p. 447)": "string"
    }
  }
},
"adMarkers (p. 449)": [enum]
"programDateTimePeriod (p. 450)": integer,
"manifestCompression (p. 450)": enum,
"segmentLength (p. 450)": integer,
"manifestDurationFormat (p. 450)": enum
"
},
"name (p. 490)": "string",
"customName (p. 490)": "string"
"},
"nielsenConfiguration (p. 460)": {
"distributorId (p. 486)": "string",
"breakoutCode (p. 486)": integer
"},
"inputs (p. 461)": [
{ "audioSelectors (p. 453)": {
 },
"audioSelectorGroups (p. 453)": {
 },
"videoSelector (p. 453)": {
 "colorSpace (p. 505)": enum,
 "hdr10Metadata (p. 505)": {
 "redPrimaryY (p. 442)": integer,
 "greenPrimaryY (p. 443)": integer,
 "whitePointX (p. 443)": integer,
 "maxLuminance (p. 443)": integer,
 "greenPrimaryX (p. 443)": integer,
 "whitePointY (p. 443)": integer,
 "redPrimaryX (p. 444)": integer,
 "bluePrimaryX (p. 444)": integer,
 "bluePrimaryY (p. 444)": integer,
 "maxFrameAverageLightLevel (p. 444)": integer,
 "maxContentLightLevel (p. 444)": integer,
 "minLuminance (p. 445)": integer
 },
 "programNumber (p. 505)": integer,
 "pid (p. 505)": integer,
 "colorSpaceUsage (p. 506)": enum
 },
 "denoiseFilter (p. 453)": enum,
 "filterEnable (p. 453)": enum,
 "deblockFilter (p. 453)": enum,
 "filterStrength (p. 453)": integer,
 "programNumber (p. 454)": integer,
 "timecodeSource (p. 454)": enum,
 "captionSelectors (p. 454)": {
 },
 "fileInput (p. 454)": "string",
 "inputClippings (p. 454)": [
 { "startTimecode (p. 455)": "string",
 "endTimecode (p. 455)": "string"
 },
 "psiControl (p. 455)": enum
 },
 "timedMetadataInsertion (p. 461)": {
 "id3Insertions (p. 500)": [
 { "id3 (p. 452)": "string",
 "timecode (p. 452)": "string"
 ]
 ]}.
### Schemas

#### "availBlanking (p. 461)"
- "availBlankingImage (p. 394)" : "string"

#### "role (p. 408)"
- "ClientRequestToken (p. 408)" : "string"
- "JobTemplate (p. 408)" : "string"
- "UserMetadata (p. 408)" : {
- "queue (p. 408)" : "string"

### Response Bodies

**Example ListJobsResponse**

```json
{
  "nextToken (p. 465)" : "string",
  "jobs (p. 466)" : [
    {
      "settings (p. 458)" : {
        "timecodeConfig (p. 460)" : {
          "timestampOffset (p. 499)" : "string",
          "anchor (p. 499)" : "string",
          "start (p. 499)" : "string",
          "source (p. 500)" : "enum",
        },
        "adAvailOffset (p. 460)" : integer,
        "outputGroups (p. 460)" : [
          {
            "outputs (p. 490)" : [
              {
                "extension (p. 488)" : "string",
                "videoDescription (p. 488)" : {
                  "fixedAfd (p. 502)" : integer,
                  "scalingBehavior (p. 502)" : "enum",
                  "respondToAfd (p. 502)" : "enum",
                  "codecSettings (p. 502)" : {
                    "h265Settings (p. 501)" : {
                      "slices (p. 436)" : integer,
                      "minIInterval (p. 437)" : integer,
                      "parNumerator (p. 437)" : integer,
                      "flickerAdaptiveQuantization (p. 437)" : "enum",
                      "gopSizeUnits (p. 437)" : "enum",
                      "hrdBufferSize (p. 437)" : integer,
                      "qualityTuningLevel (p. 437)" : "enum",
                      "maxBitrate (p. 437)" : integer,
                      "bitrate (p. 438)" : integer,
                      "spatialAdaptiveQuantization (p. 438)" : "enum",
                      "sampleAdaptiveOffsetFilterMode (p. 438)" : "enum",
                      "temporalIds (p. 438)" : "enum",
                      "slowPal (p. 438)" : "enum",
                      "tiles (p. 438)" : "enum",
                      "writeMp4PackagingType (p. 438)" : "enum",
                      "codecProfile (p. 438)" : "enum",
                      "alternateTransferFunctionSei (p. 438)" : "enum",
                      "unregisteredSeiTimecode (p. 439)" : "enum",
                      "framerateControl (p. 439)" : "enum",
                      "telecine (p. 439)" : "enum",
                      "codecLevel (p. 439)" : "enum",
                      "framerateConversionAlgorithm (p. 439)" : "enum",
                      "numberReferenceFrames (p. 439)" : integer,
                      "temporalAdaptiveQuantization (p. 439)" : "enum",
                      "hrdBufferInitialFillPercentage (p. 439)" : integer,
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  ]
}
```
"framerateNumerator (p. 439)": integer,
"gopClosedCadence (p. 440)": integer,
"numberBFramesBetweenReferenceFrames (p. 440)": integer,
"framerateDenominator (p. 440)": integer,
"adaptiveQuantization (p. 440)": enum,
"interlaceMode (p. 440)": enum,
"gopSize (p. 440)": number,
"gopBReference (p. 441)": enum,
"parDenominator (p. 441)": integer,
"sceneChangeDetect (p. 441)": enum,
"parControl (p. 441)": enum,
"rateControlMode (p. 441)": enum
},
"codec (p. 501)": enum,
"proresSettings (p. 501)": {
  "framerateDenominator (p. 493)": integer,
  "slowPal (p. 493)": enum,
  "framerateControl (p. 493)": enum,
  "telecine (p. 493)": enum,
  "framerateConversionAlgorithm (p. 493)": enum,
  "interlaceMode (p. 493)": enum,
  "parNumerator (p. 493)": integer,
  "codecProfile (p. 493)": enum,
  "parDenominator (p. 494)": integer,
  "parControl (p. 494)": enum,
  "framerateNumerator (p. 494)": integer
},
"mpeg2Settings (p. 502)": {
  "minInterval (p. 480)": integer,
  "parNumerator (p. 480)": integer,
  "gopSizeUnits (p. 481)": enum,
  "hrdBufferSize (p. 481)": integer,
  "qualityTuningLevel (p. 481)": enum,
  "maxBitrate (p. 481)": integer,
  "bitrate (p. 481)": integer,
  "spatialAdaptiveQuantization (p. 481)": enum,
  "slowPal (p. 481)": enum,
  "intraDcPrecision (p. 482)": enum,
  "codecProfile (p. 482)": enum,
  "softness (p. 482)": integer,
  "framerateControl (p. 482)": enum,
  "telecine (p. 482)": enum,
  "codecLevel (p. 482)": enum,
  "framerateConversionAlgorithm (p. 482)": enum,
  "temporalAdaptiveQuantization (p. 482)": enum,
  "hrdBufferInitialFillPercentage (p. 482)": integer,
  "framerateNumerator (p. 483)": integer,
  "gopClosedCadence (p. 483)": integer,
  "numberBFramesBetweenReferenceFrames (p. 483)": integer,
  "framerateDenominator (p. 483)": integer,
  "adaptiveQuantization (p. 483)": enum,
  "interlaceMode (p. 483)": enum,
  "gopSize (p. 483)": number,
  "syntax (p. 484)": enum,
  "parDenominator (p. 484)": integer,
  "sceneChangeDetect (p. 484)": enum,
  "parControl (p. 484)": enum,
  "rateControlMode (p. 484)": enum
},
"frameCaptureSettings (p. 502)": {
  "framerateDenominator (p. 424)": integer,
  "maxCaptures (p. 424)": integer,
  "framerateNumerator (p. 424)": integer,
  "quality (p. 424)": integer
},
"h264Settings (p. 502)": {

"slices (p. 428)": integer,
"minIInterval (p. 428)": integer,
"parNumerator (p. 428)": integer,
"flickerAdaptiveQuantization (p. 428)": enum,
"gopSizeUnits (p. 428)": enum,
"hrdBufferSize (p. 428)": integer,
"qualityTuningLevel (p. 429)": enum,
"maxBitrate (p. 429)": integer,
"bitrate (p. 429)": integer,
"spatialAdaptiveQuantization (p. 429)": enum,
"slowPal (p. 429)": enum,
"codecProfile (p. 429)": enum,
"unregisteredSeiTimecode (p. 429)": enum,
"softness (p. 429)": integer,
"framerateControl (p. 430)": enum,
"telecine (p. 430)": enum,
"codecLevel (p. 430)": enum,
"framerateConversionAlgorithm (p. 430)": enum,
"numberReferenceFrames (p. 430)": integer,
"temporalAdaptiveQuantization (p. 430)": enum,
"repeatPps (p. 430)": enum,
"hrdBufferInitialFillPercentage (p. 430)": integer,
"framerateNumerator (p. 430)": integer,
"gopClosedCadence (p. 431)": integer,
"numberBFramesBetweenReferenceFrames (p. 431)": integer,
"framerateDenominator (p. 431)": integer,
"entropyEncoding (p. 431)": enum,
"fieldEncoding (p. 431)": enum,
"adaptiveQuantization (p. 432)": enum,
"interlaceMode (p. 432)": enum,
"gopSize (p. 432)": number,
"gopBReference (p. 432)": enum,
"syntax (p. 432)": enum,
"parDenominator (p. 432)": integer,
"sceneChangeDetect (p. 432)": enum,
"parControl (p. 432)": enum,
"rateControlMode (p. 432)": enum
},
"afdSignaling (p. 502)": enum,
"colorMetadata (p. 502)": enum,
"timecodeInsertion (p. 503)": enum,
"width (p. 503)": integer,
"videoPreprocessors (p. 503)": {
  "noiseReducer (p. 504)": {
    "filter (p. 486)": enum,
    "filterSettings (p. 486)": {
      "strength (p. 487)": integer
    },
    "spatialFilterSettings (p. 487)": {
      "strength (p. 487)": integer,
      "postFilterSharpenStrength (p. 487)": integer,
      "speed (p. 488)": integer
    }
  },
  "timecodeBurnin (p. 504)": {
    "prefix (p. 498)": "string",
    "fontSize (p. 498)": integer,
    "position (p. 498)": enum
  },
  "colorCorrector (p. 505)": {
    "saturation (p. 405)": integer,
    "brightness (p. 405)": integer,
    "hdr10Metadata (p. 405)": {
      "redPrimaryY (p. 442)": integer,
      "greenPrimaryY (p. 443)": integer,
      "
"whitePointX (p. 443)": integer,
"maxLuminance (p. 443)": integer,
"greenPrimaryX (p. 443)": integer,
"whitePointY (p. 443)": integer,
"redPrimaryX (p. 444)": integer,
"bluePrimaryX (p. 444)": integer,
"bluePrimaryY (p. 444)": integer,
"maxFrameAverageLightLevel (p. 444)": integer,
"maxContentLightLevel (p. 444)": integer,
"minLuminance (p. 445)": integer
},
"contrast (p. 405)": integer,
"hue (p. 405)": integer,
"colorSpaceConversion (p. 406)": enum
},
"deinterlacer (p. 505)": {
"mode (p. 411)": enum,
"control (p. 411)": enum,
"algorithm (p. 411)": enum
},
"imageInserter (p. 505)": {
"insertableImages (p. 453)": [ {
"duration (p. 456)": integer,
"fadeOut (p. 456)": integer,
"imageY (p. 457)": integer,
"fadeIn (p. 457)": integer,
"imageX (p. 457)": integer,
"width (p. 457)": integer,
"startTime (p. 457)": "string",
"opacity (p. 457)": integer,
"layer (p. 458)": integer,
"height (p. 458)": integer,
"imageInserterInput (p. 458)": "string"
} ]
},
"antiAlias (p. 503)": enum,
"position (p. 503)": {
"width (p. 494)": integer,
"x (p. 495)": integer,
"y (p. 495)": integer,
"height (p. 495)": integer
},
"sharpness (p. 503)": integer,
"crop (p. 503)": {
"width (p. 494)": integer,
"x (p. 495)": integer,
"y (p. 495)": integer,
"height (p. 495)": integer
},
"dropFrameTimecode (p. 504)": enum,
"height (p. 504)": integer
},
"audioDescriptions (p. 488)": [ {
"audioTypeControl (p. 391)": enum,
"languageCodeControl (p. 391)": enum,
"remixSettings (p. 391)": {
"channelsOut (p. 495)": integer,
"channelMapping (p. 495)": {
"outputChannels (p. 401)": [ { "inputChannels (p. 489)": [ integer

366
"audioSourceName (p. 391)": string,
"codecSettings (p. 392)": {
  "codec (p. 390)": enum,
  "wavSettings (p. 390)": {
    "channels (p. 506)": integer,
    "bitDepth (p. 506)": integer,
    "format (p. 506)": enum,
    "sampleRate (p. 507)": integer
  },
  "aacSettings (p. 390)": {
    "vbrQuality (p. 385)": enum,
    "codecProfile (p. 385)": enum,
    "codingMode (p. 385)": enum,
    "specification (p. 386)": enum,
    "bitrate (p. 386)": integer,
    "rawFormat (p. 386)": enum,
    "rateControlMode (p. 386)": enum,
    "sampleRate (p. 386)": integer,
    "audioDescriptionBroadcasterMix (p. 386)": enum
  },
  "ac3Settings (p. 390)": {
    "dynamicRangeCompressionProfile (p. 387)": enum,
    "dialnorm (p. 388)": integer,
    "codingMode (p. 388)": enum,
    "metadataControl (p. 388)": enum,
    "bitrate (p. 388)": integer,
    "lfeFilter (p. 388)": enum,
    "bitstreamMode (p. 388)": enum,
    "sampleRate (p. 388)": integer
  },
  "aiffSettings (p. 390)": {
    "channels (p. 389)": integer,
    "bitDepth (p. 389)": integer,
    "sampleRate (p. 389)": integer
  },
  "eac3Settings (p. 390)": {
    "dialnorm (p. 419)": integer,
    "passsthroughControl (p. 419)": enum,
    "metadataControl (p. 419)": enum,
    "bitrate (p. 419)": integer,
    "dynamicRangeCompressionProfile (p. 419)": enum,
    "sampleRate (p. 419)": integer,
    "surroundExMode (p. 419)": enum,
    "l1rtSurroundMixLevel (p. 420)": number,
    "dynamicRangeCompressionLine (p. 420)": enum,
    "lfeControl (p. 420)": enum,
    "codingMode (p. 420)": enum,
    "surroundMode (p. 420)": enum,
    "attenuationControl (p. 420)": enum,
    "lfeFilter (p. 420)": enum,
    "phaseControl (p. 420)": enum,
    "l1rCenterMixLevel (p. 420)": number,
    "dcFilter (p. 421)": enum,
    "bitstreamMode (p. 421)": enum,
    "stereoDownmix (p. 421)": enum,
    "loRoSurroundMixLevel (p. 421)": number,
    "loRoCenterMixLevel (p. 421)": number
  },
  "mp2Settings (p. 390)": {
    "channelsIn (p. 495)": integer
  }
}
"channels (p. 477)": integer,
"bitrate (p. 477)": integer,
"sampleRate (p. 477)": integer
},
"languageCode (p. 392)": enum,
"streamName (p. 392)": "string",
"audioNormalizationSettings (p. 392)": {
  "targetLkfs (p. 393)": number,
  "algorithmControl (p. 393)": enum,
  "loudnessLogging (p. 393)": enum,
  "peakCalculation (p. 393)": enum,
  "correctionGateLevel (p. 394)": integer,
  "algorithm (p. 394)": enum
},
"customLanguageCode (p. 392)": "string"
},
"containerSettings (p. 488)": {
  "container (p. 407)": enum,
  "mp4Settings (p. 407)": {
    "mp4MajorBrand (p. 478)": "string",
    "moovPlacement (p. 478)": enum,
    "cslgAtom (p. 478)": enum,
    "freeSpaceBox (p. 478)": enum
  },
  "m3u8Settings (p. 407)": {
    "pmtPid (p. 473)": integer,
    "nielsenId3 (p. 473)": enum,
    "pcrPid (p. 473)": integer,
    "audioPids (p. 473)": [integer
    ],
    "audioFramesPerPes (p. 473)": integer,
    "scte35Pid (p. 474)": integer,
    "transportStreamId (p. 474)": integer,
    "pcrControl (p. 474)": enum,
    "videoPid (p. 474)": integer,
    "privateMetadataPid (p. 474)": integer,
    "pmtInterval (p. 474)": integer,
    "patInterval (p. 475)": integer,
    "programNumber (p. 475)": integer,
    "timedMetadataPid (p. 475)": integer,
    "timedMetadata (p. 475)": enum,
    "scte35Source (p. 475)": enum
  },
  "m2tsSettings (p. 407)": {
    "dvbTeletextPid (p. 468)": integer,
    "nielsenId3 (p. 468)": enum,
    "bitrate (p. 468)": integer,
    "segmentationTime (p. 468)": number,
    "audioPids (p. 468)": [integer
    ],
    "rateMode (p. 468)": enum,
    "maxPcrInterval (p. 468)": integer,
    "audioFramesPerPes (p. 469)": integer,
    "ebpAudioInterval (p. 469)": enum,
    "fragmentTime (p. 469)": number,
    "scte35Pid (p. 469)": integer,
    "pmtInterval (p. 469)": integer,
    "privateMetadataPid (p. 469)": integer,
    "segmentationStyle (p. 470)": enum,
    "audioBufferModel (p. 470)": enum,
    "programNumber (p. 470)": integer,
    "dvbNitSettings (p. 470)": {
      "dvbNitPid (p. 470)": integer,
      "dvbNitTs (p. 470)": integer,
      "dvbNitInterval (p. 470)": integer
    }
  }
}
"networkName (p. 411)": "string",
"networkId (p. 412)": integer,
"nitInterval (p. 412)": integer
},
"timedMetadataPid (p. 470)": integer,
"scte35Source (p. 470)": enum,
"pmtPid (p. 470)": integer,
"bufferModel (p. 470)": enum,
"ebpPlacement (p. 471)": enum,
"dvbSdtSettings (p. 471)": {
  "sdInterval (p. 412)": integer,
  "serviceName (p. 412)": "string",
  "serviceProviderName (p. 412)": "string",
  "outputSdt (p. 412)": enum
},
"nullPacketBitrate (p. 471)": number,
"pcrPid (p. 471)": integer,
"minEbpInterval (p. 471)": integer,
"transportStreamId (p. 471)": integer,
"videoPid (p. 472)": integer,
"pcrControl (p. 472)": enum,
"esRateInPes (p. 472)": enum,
"segmentationMarkers (p. 472)": enum,
"dvbTdtSettings (p. 472)": {
  "tdtInterval (p. 417)": integer
},
"patInterval (p. 472)": integer,
"dvbSubPids (p. 472)": [integer]
},
"movSettings (p. 407)": {
  "reference (p. 476)": enum,
  "paddingControl (p. 476)": enum,
  "mpeg2FourCCControl (p. 476)": enum,
  "csigAtom (p. 476)": enum,
  "clapAtom (p. 476)": enum
},
"f4vSettings (p. 407)": {
  "moovPlacement (p. 423)": enum
},
"preset (p. 489)": "string",
"outputSettings (p. 489)": {
  "hlsSettings (p. 491)": {
    "iFrameOnlyManifest (p. 451)": enum,
    "audioRenditionSets (p. 451)": "string",
    "audioTrackType (p. 452)": enum,
    "segmentModifier (p. 452)": "string",
    "audioGroupId (p. 452)": "string"
  }
},
"captionDescriptions (p. 489)": [
  {
    "captionSelectorName (p. 398)": "string",
    "languageDescription (p. 398)": "string",
    "destinationSettings (p. 399)": {
      "burnInDestinationSettings (p. 399)": {
        "xPosition (p. 394)": integer,
        "backgroundColor (p. 395)": enum,
        "teletextSpacing (p. 395)": enum,
        "yPosition (p. 395)": integer,
        "backgroundOpacity (p. 395)": integer,
        "fontOpacity (p. 395)": integer,
        "fontResolution (p. 395)": integer,
        "shadowOpacity (p. 396)": integer,
"shadowYOffset (p. 396)" : integer,
"outlineSize (p. 396)" : integer,
"outlineColor (p. 396)" : enum,
"fontSize (p. 396)" : integer,
"shadowXOffset (p. 396)" : integer,
"alignment (p. 397)" : enum,
"shadowColor (p. 397)" : enum,
"fontColor (p. 397)" : enum
},
"teletextDestinationSettings (p. 399)" : {
  "pageNumber (p. 498)" : "string"
},
"ttmlDestinationSettings (p. 399)" : {
  "stylePassthrough (p. 501)" : enum
},
"destinationType (p. 399)" : enum,
"dvbSubDestinationSettings (p. 399)" : {
  "xPosition (p. 413)" : integer,
  "backgroundColor (p. 413)" : enum,
  "teletextSpacing (p. 413)" : enum,
  "yPosition (p. 413)" : integer,
  "backgroundOpacity (p. 413)" : integer,
  "fontOpacity (p. 413)" : integer,
  "fontResolution (p. 414)" : integer,
  "shadowOpacity (p. 414)" : integer,
  "shadowYOffset (p. 414)" : integer,
  "outlineSize (p. 414)" : integer,
  "outlineColor (p. 414)" : enum,
  "fontSize (p. 414)" : integer,
  "shadowXOffset (p. 415)" : integer,
  "alignment (p. 415)" : enum,
  "shadowColor (p. 415)" : enum,
  "fontColor (p. 415)" : enum
},
"sccDestinationSettings (p. 399)" : {
  "framerate (p. 496)" : enum
},
"languageCode (p. 399)" : enum,
"customLanguageCode (p. 399)" : "string"
}],
"nameModifier (p. 489)" : "string"
},
"outputGroupSettings (p. 490)" : {
  "dashIsoGroupSettings (p. 490)" : {
    "fragmentLength (p. 409)" : integer,
    "baseUrl (p. 409)" : "string",
    "minBufferTime (p. 409)" : integer,
    "encryption (p. 409)" : {
      "spekeKeyProvider (p. 409)" : {
        "resourceId (p. 496)" : "string",
        "systemIds (p. 497)" : [
          "string"
        ],
        "url (p. 497)" : "string"
      }
    },
    "destination (p. 409)" : "string",
    "segmentLength (p. 410)" : integer,
    "segmentControl (p. 410)" : enum,
    "hbbtvCompliance (p. 410)" : enum
  },
  "fileGroupSettings (p. 490)" : {
    "destination (p. 423)" : "string"
  }
}
"msSmoothGroupSettings (p. 491)": {
    "fragmentLength (p. 485)": integer,
    "encryption (p. 485)": {
        "spekeKeyProvider (p. 485)": {
            "resourceId (p. 496)": "string",
            "systemIds (p. 497)": ["string"
            ],
            "url (p. 497)": "string"
        }},
    "audioDeduplication (p. 485)": enum,
    "manifestEncoding (p. 486)": enum,
    "destination (p. 486)": "string"
},
"cmafGroupSettings (p. 491)": {
    "writeHlsManifest (p. 402)": enum,
    "writeDashManifest (p. 402)": enum,
    "fragmentLength (p. 402)": integer,
    "streamInfResolution (p. 402)": enum,
    "minBufferTime (p. 402)": integer,
    "clientCache (p. 403)": enum,
    "codecSpecification (p. 403)": enum,
    "destination (p. 403)": "string",
    "segmentControl (p. 403)": enum,
    "baseUrl (p. 403)": "string",
    "encryption (p. 403)": {
        "initializationVectorInManifest (p. 401)": enum,
        "constantInitializationVector (p. 401)": "string",
        "staticKeyProvider (p. 401)": {
            "keyFormatVersions (p. 497)": "string",
            "keyFormat (p. 497)": "string",
            "staticKeyValue (p. 497)": "string",
            "url (p. 497)": "string"
        },
        "type (p. 402)": enum,
        "encryptionMethod (p. 402)": enum
    },
    "manifestCompression (p. 403)": enum,
    "segmentLength (p. 403)": integer,
    "manifestDurationFormat (p. 404)": enum
},
"type (p. 491)": enum,
"hlsGroupSettings (p. 491)": {
    "segmentsPerSubdirectory (p. 447)": integer,
    "streamInfResolution (p. 447)": enum,
    "timestampDeltaMilliseconds (p. 448)": integer,
    "outputSelection (p. 448)": enum,
    "captionLanguageMappings (p. 448)": [
        {"languageDescription (p. 445)": "string",
        "captionChannel (p. 445)": integer,
        "languageCode (p. 445)": enum,
        "customLanguageCode (p. 446)": "string"
        }
    ],
    "clientCache (p. 448)": enum,
    "codecSpecification (p. 448)": enum,
    "destination (p. 448)": "string",
    "timedMetadataId3Frame (p. 448)": enum,
    "segmentControl (p. 448)": enum,
    "timedMetadataId3Period (p. 449)": integer,
    "captionLanguageSetting (p. 449)": enum,
    "minSegmentLength (p. 449)": integer,
    "directoryStructure (p. 449)": enum
}
Schemas

"programDateTime (p. 449)": enum,
"baseUrl (p. 449)": "string",
"encryption (p. 449)": {
  "initializationVectorInManifest (p. 446)": enum,
  "constantInitializationVector (p. 447)": "string",
  "staticKeyProvider (p. 447)": {
    "keyFormatVersions (p. 497)": "string",
    "keyFormat (p. 497)": "string",
    "staticKeyValue (p. 497)": "string",
    "url (p. 497)": "string"
  },
  "type (p. 447)": enum,
  "encryptionMethod (p. 447)": enum,
  "spekeKeyProvider (p. 447)": {
    "resourceId (p. 496)": "string",
    "systemIds (p. 497)": ["string"],
    "url (p. 497)": "string"
  },
  "url (p. 497)": "string"
},
"adMarkers (p. 449)": [enum],
"programDateTimePeriod (p. 450)": integer,
"manifestCompression (p. 450)": enum,
"segmentLength (p. 450)": integer,
"manifestDurationFormat (p. 450)": enum
},
"name (p. 490)": "string",
"customName (p. 490)": "string"
],
"nielsenConfiguration (p. 460)": {
  "distributorId (p. 486)": "string",
  "breakoutCode (p. 486)": integer
},
"inputs (p. 461)": [
  {
    "audioSelectors (p. 453)": {
    },
    "audioSelectorGroups (p. 453)": {
    },
    "videoSelector (p. 453)": {
      "colorSpace (p. 505)": enum,
      "hdr10Metadata (p. 505)": {
        "redPrimaryY (p. 442)": integer,
        "greenPrimaryY (p. 443)": integer,
        "whitePointX (p. 443)": integer,
        "maxLuminance (p. 443)": integer,
        "greenPrimaryX (p. 443)": integer,
        "whitePointY (p. 443)": integer,
        "redPrimaryX (p. 444)": integer,
        "bluePrimaryX (p. 444)": integer,
        "bluePrimaryY (p. 444)": integer,
        "maxFrameAverageLightLevel (p. 444)": integer,
        "maxContentLightLevel (p. 444)": integer,
        "minLuminance (p. 445)": integer
      },
      "programNumber (p. 505)": integer,
      "pid (p. 505)": integer,
      "colorSpaceUsage (p. 506)": enum
    }
  },
"denoiseFilter (p. 453)": enum,
"filterEnable (p. 453)": enum,
"deblockFilter (p. 453)": enum,
"filterStrength (p. 453)": integer,
"programNumber (p. 454)": integer,
"timecodeSource (p. 454)": enum,
"captionSelectors (p. 454)": {
},
"fileInput (p. 454)": "string",
"inputClippings (p. 454)": [
  {
    "startTimecode (p. 455)": "string",
    "endTimecode (p. 455)": "string"
  }
],
"psiControl (p. 455)": enum
],
"timedMetadataInsertion (p. 461)": {
  "id3Insertions (p. 500)": [
    {
      "id3 (p. 452)": "string",
      "timecode (p. 452)": "string"
    }
  ],
  "availBlanking (p. 461)": {
    "availBlankingImage (p. 394)": "string"
  },
  "role (p. 458)": "string",
  "outputGroupDetails (p. 458)": [
    {
      "outputDetails (p. 490)": [
        {
          "durationInMs (p. 489)": integer,
          "videoDetails (p. 489)": {
            "heightInPx (p. 504)": integer,
            "widthInPx (p. 504)": integer
          }
        }
      ],
      "jobTemplate (p. 459)": "string",
      "timing (p. 459)": {
        "finishTime (p. 500)": "string",
        "submitTime (p. 500)": "string",
        "startTime (p. 501)": "string"
      },
      "errorMessage (p. 459)": "string",
      "errorCode (p. 459)": integer,
      "createdAt (p. 459)": "string",
      "id (p. 459)": "string",
      "arn (p. 459)": "string",
      "userMetadata (p. 459)": {
      },
      "queue (p. 460)": "string",
      "status (p. 460)": enum
    }
  ]
]
"job (p. 408)": {
  "settings (p. 458)": {
    "timecodeConfig (p. 460)": {
      "timestampOffset (p. 499)": "string",
      "anchor (p. 499)": "string",
      "start (p. 499)": "string",
      "source (p. 500)": enum
    },
    "adAvailableOffset (p. 460)": integer,
    "outputGroups (p. 460)": {
      "outputs (p. 490)": [
        {
          "extension (p. 488)": "string",
          "videoDescription (p. 488)": {
            "fixedAfd (p. 502)": integer,
            "scalingBehavior (p. 502)": enum,
            "respondToAfd (p. 502)": enum,
            "codecSettings (p. 502)": {
              "h265Settings (p. 501)": {
                "slices (p. 436)": integer,
                "minIInterval (p. 437)": integer,
                "parNumerator (p. 437)": integer,
                "flickerAdaptiveQuantization (p. 437)": enum,
                "gopSizeUnits (p. 437)": enum,
                "hrdBufferSize (p. 437)": integer,
                "qualityTuningLevel (p. 437)": enum,
                "maxBitrate (p. 437)": integer,
                "bitrate (p. 438)": integer,
                "spatialAdaptiveQuantization (p. 438)": enum,
                "sampleAdaptiveOffsetFilterMode (p. 438)": enum,
                "temporalIds (p. 438)": enum,
                "slowPal (p. 438)": enum,
                "tiles (p. 438)": enum,
                "writeMp4PackagingType (p. 438)": enum,
                "codecProfile (p. 438)": enum,
                "alternateTransferFunctionSei (p. 438)": enum,
                "framerateControl (p. 439)": enum,
                "telecine (p. 439)": enum,
                "framerateConversionAlgorithm (p. 439)": enum,
                "unregisteredSeiTimecode (p. 439)": enum,
                "framerateNumerator (p. 439)": integer,
                "gopClosedCadence (p. 440)": integer,
                "framerateDenominator (p. 440)": integer,
                "framerateBReference (p. 440)": enum,
                "gopSize (p. 440)": number,
                "gopBitrate (p. 441)": enum,
                "parDenominator (p. 441)": integer,
                "sceneChangeDetect (p. 441)": enum,
                "parControl (p. 441)": enum,
                "rateControlMode (p. 441)": enum
              },
              "proresSettings (p. 501)": {
                "framerateDenominator (p. 493)": integer,
                "slowPal (p. 493)": enum,
                "framerateControl (p. 493)": enum,
                "telecine (p. 493)": enum,
                "framerateConversionAlgorithm (p. 493)": enum,
                "interlaceMode (p. 493)": enum
              }
            }
          }
        }
      }
    }
  }
}
"parNumerator (p. 493)" : integer,
"codecProfile (p. 493)" : enum,
"parDenominator (p. 494)" : integer,
"parControl (p. 494)" : enum,
"framerateNumerator (p. 494)" : integer
},
"mpeg2Settings (p. 502)" : {
  "minIInterval (p. 480)" : integer,
  "parNumerator (p. 480)" : integer,
  "gopSizeUnits (p. 481)" : enum,
  "hrdBufferSize (p. 481)" : integer,
  "qualityTuningLevel (p. 481)" : enum,
  "maxBitrate (p. 481)" : integer,
  " bitrate (p. 481)" : integer,
  "spatialAdaptiveQuantization (p. 481)" : enum,
  "slowPal (p. 481)" : enum,
  "intraDCPrecision (p. 482)" : enum,
  "codecProfile (p. 482)" : enum,
  "softness (p. 482)" : integer,
  "framerateControl (p. 482)" : enum,
  "telecine (p. 482)" : enum,
  "codecLevel (p. 482)" : enum,
  "framerateConversionAlgorithm (p. 482)" : enum,
  "temporalAdaptiveQuantization (p. 482)" : enum,
  "framerateClosedCadence (p. 483)" : integer,
  "framerateDenominator (p. 483)" : integer,
  "adaptiveQuantization (p. 483)" : enum,
  "interlaceMode (p. 483)" : enum,
  "gopSize (p. 483)" : number,
  "syntax (p. 484)" : enum,
  "parDenominator (p. 484)" : integer,
  "sceneChangeDetect (p. 484)" : enum,
  "framerateControl (p. 484)" : enum,
  "rateControlMode (p. 484)" : enum
},
"frameCaptureSettings (p. 502)" : {
  "framerateDenominator (p. 424)" : integer,
  "maxCaptures (p. 424)" : integer,
  "framerateNumerator (p. 424)" : integer,
  "quality (p. 424)" : integer
},
"h264Settings (p. 502)" : {
  "slices (p. 428)" : integer,
  "minIInterval (p. 428)" : integer,
  "parNumerator (p. 428)" : integer,
  "flickerAdaptiveQuantization (p. 428)" : enum,
  "gopSizeUnits (p. 428)" : enum,
  "hrdBufferSize (p. 428)" : integer,
  "qualityTuningLevel (p. 429)" : enum,
  "maxBitrate (p. 429)" : integer,
  " bitrate (p. 429)" : integer,
  "spatialAdaptiveQuantization (p. 429)" : enum,
  "slowPal (p. 429)" : enum,
  "codecProfile (p. 429)" : enum,
  "unregisteredSeiTimecode (p. 429)" : enum,
  "softness (p. 429)" : integer,
  "framerateControl (p. 430)" : enum,
  "telecine (p. 430)" : enum,
  "codecLevel (p. 430)" : enum,
  "framerateConversionAlgorithm (p. 430)" : enum,
  "numberReferenceFrames (p. 430)" : integer,
  "temporalAdaptiveQuantization (p. 430)" : enum,
  "repeatPps (p. 430)" : enum,
"hrdBufferInitialFillPercentage (p. 430)": integer,
"framerateNumerator (p. 431)": integer,
"gopClosedCadence (p. 431)": integer,
"numberBFramesBetweenReferenceFrames (p. 431)": integer,
"framerateDenominator (p. 431)": integer,
"entropyEncoding (p. 431)": enum,
"fieldEncoding (p. 431)": enum,
"adaptiveQuantization (p. 432)": enum,
"interlaceMode (p. 432)": enum,
"gopSize (p. 432)": number,
"gopBReference (p. 432)": enum,
"syntax (p. 432)": enum,
"parDenominator (p. 432)": integer,
"sceneChangeDetect (p. 432)": enum,
"parControl (p. 432)": enum,
"rateControlMode (p. 432)": enum
}

"afdsignaling (p. 502)": enum,
"colorMetadata (p. 502)": enum,
"timecodeInsertion (p. 503)": enum,
"width (p. 503)": integer,
"videoPreprocessors (p. 503)": {
  "noiseReducer (p. 504)": {
    "filter (p. 486)": enum,
    "filterSettings (p. 486)": {
      "strength (p. 487)": integer
    },
    "spatialFilterSettings (p. 487)": {
      "strength (p. 487)": integer,
      "postFilterSharpenStrength (p. 487)": integer,
      "speed (p. 488)": integer
    }
  },
  "timecodeBurnin (p. 504)": {
    "prefix (p. 498)": "string",
    "fontSize (p. 498)": integer,
    "position (p. 498)": enum
  },
  "colorCorrector (p. 505)": {
    "saturation (p. 405)": integer,
    "brightness (p. 405)": integer,
    "hdr10Metadata (p. 405)": {
      "redPrimaryY (p. 442)": integer,
      "greenPrimaryY (p. 443)": integer,
      "whitePointX (p. 443)": integer,
      "maxLuminance (p. 443)": integer,
      "redPrimaryX (p. 444)": integer,
      "bluePrimaryX (p. 444)": integer,
      "bluePrimaryY (p. 444)": integer,
      "maxFrameAverageLightLevel (p. 444)": integer,
      "maxContentLightLevel (p. 444)": integer,
      "minLuminance (p. 445)": integer
    },
    "contrast (p. 405)": integer,
    "hue (p. 405)": integer,
    "colorSpaceConversion (p. 406)": enum
  },
  "deinterlacer (p. 505)": {
    "mode (p. 411)": enum,
    "control (p. 411)": enum,
    "algorithm (p. 411)": enum
  },
  "imageInserter (p. 505)": {
"
"insertableImages (p. 453)": [  
  {  
    "duration (p. 456)": integer,  
    "fadeOut (p. 456)": integer,  
    "imageY (p. 457)": integer,  
    "fadeIn (p. 457)": integer,  
    "imageX (p. 457)": integer,  
    "width (p. 457)": integer,  
    "startTime (p. 457)": "string",  
    "opacity (p. 457)": integer,  
    "layer (p. 458)": integer,  
    "height (p. 458)": integer,  
    "imageInserterInput (p. 458)": "string"  
  }  
],  
"antiAlias (p. 503)": enum,  
"position (p. 503)": {  
  "width (p. 494)": integer,  
  "x (p. 495)": integer,  
  "y (p. 495)": integer,  
  "height (p. 495)": integer  
},  
"sharpness (p. 503)": integer,  
"crop (p. 503)": {  
  "width (p. 494)": integer,  
  "x (p. 495)": integer,  
  "y (p. 495)": integer,  
  "height (p. 495)": integer  
},  
"dropFrameTimecode (p. 504)": enum,  
"height (p. 504)": integer  
],  
"audioDescriptions (p. 488)": [  
  {  
    "audioTypeControl (p. 391)": enum,  
    "languageCodeControl (p. 391)": enum,  
    "remixSettings (p. 391)": {  
      "channelsOut (p. 495)": integer,  
      "channelMapping (p. 495)": {  
        "outputChannels (p. 401)": [  
          "inputChannels (p. 409)": [  
            integer  
          ]  
        ]  
      }  
    },  
    "channelsIn (p. 495)": integer  
  },  
  "audioType (p. 391)": integer,  
  "audioSourceName (p. 391)": "string",  
  "codecSettings (p. 392)": {  
    "codec (p. 390)": enum,  
    "wavSettings (p. 390)": {  
      "channels (p. 506)": integer,  
      "bitDepth (p. 506)": integer,  
      "format (p. 506)": enum,  
      "sampleRate (p. 507)": integer  
    },  
    "aacSettings (p. 390)": {  
      "vbrQuality (p. 385)": enum,  
      "codecProfile (p. 385)": enum,  
      "codingMode (p. 385)": enum,  
      "specification (p. 386)": enum,  
      "specification (p. 386)": enum  
    }  
  }  
]
"bitrate (p. 386)": integer,
"rawFormat (p. 386)": enum,
"rateControlMode (p. 386)": enum,
"sampleRate (p. 386)": integer,
"audioDescriptionBroadcasterMix (p. 386)": enum
},
"ac3Settings (p. 390)": {
"dynamicRangeCompressionProfile (p. 387)": enum,
"dialnorm (p. 388)": integer,
"codingMode (p. 388)": enum,
"metadataControl (p. 388)": enum,
"bitrate (p. 388)": integer,
"lfeFilter (p. 388)": enum,
"bitstreamMode (p. 388)": enum,
"sampleRate (p. 388)": integer
},
"aiffSettings (p. 390)": {
"channels (p. 389)": integer,
"bitDepth (p. 389)": integer,
"sampleRate (p. 389)": integer
},
"eac3Settings (p. 390)": {
"dialnorm (p. 419)": integer,
"passthroughControl (p. 419)": enum,
"metadataControl (p. 419)": enum,
"bitrate (p. 419)": integer,
"dynamicRangeCompressionRf (p. 419)": enum,
"sampleRate (p. 419)": integer,
"surroundExMode (p. 419)": enum,
"ltRtSurroundMixLevel (p. 420)": number,
"dynamicRangeCompressionLine (p. 420)": enum,
"lifeControl (p. 420)": enum,
"codingMode (p. 420)": enum,
"surroundMode (p. 420)": enum,
"attenuationControl (p. 420)": enum,
"lifeFilter (p. 420)": enum,
"phaseControl (p. 420)": enum,
"ltRtCenterMixLevel (p. 420)": number,
"dcFilter (p. 421)": enum,
"bitstreamMode (p. 421)": enum,
"stereoDownmix (p. 421)": enum,
"loRoSurroundMixLevel (p. 421)": number,
"loRoCenterMixLevel (p. 421)": number
},
"mp2Settings (p. 390)": {
"channels (p. 477)": integer,
"bitrate (p. 477)": integer,
"sampleRate (p. 477)": integer
},
"languageCode (p. 392)": enum,
"streamName (p. 392)": "string",
"audioNormalizationSettings (p. 392)": {
"targetLkfs (p. 393)": number,
"algorithmControl (p. 393)": enum,
"loudnessLogging (p. 393)": enum,
"peakCalculation (p. 393)": enum,
"correctionGateLevel (p. 394)": integer,
"algorithm (p. 394)": enum
},
"customLanguageCode (p. 392)": "string"
],
"containerSettings (p. 488)": {
"container (p. 407)": enum,
"mp4Settings (p. 407)": {

"mp4MajorBrand (p. 478)": "string",
"moovPlacement (p. 478)": enum,
"cslgAtom (p. 478)": enum,
"freeSpaceBox (p. 478)": enum,
"m3u8Settings (p. 407)": {
    "pmtPid (p. 473)": integer,
    "nielsenId3 (p. 473)": enum,
    "pcrPid (p. 473)": integer,
    "audioPids (p. 473)": [integer
    ],
    "audioFramesPerPes (p. 473)": integer,
    "scte35Pid (p. 474)": integer,
    "transportStreamId (p. 474)": integer,
    "pcrControl (p. 474)": enum,
    "videoPid (p. 474)": integer,
    "privateMetadataPid (p. 474)": integer,
    "pmtInterval (p. 474)": integer,
    "patInterval (p. 475)": integer,
    "programNumber (p. 475)": integer,
    "timedMetadataPid (p. 475)": integer,
    "timedMetadata (p. 475)": enum,
    "scte35Source (p. 475)": enum
},
"m2tsSettings (p. 407)": {
    "dvbTeletextPid (p. 468)": integer,
    "nielsenId3 (p. 468)": enum,
    "bitrate (p. 468)": integer,
    "segmentationTime (p. 468)": number,
    "audioPids (p. 468)": [integer
    ],
    "rateMode (p. 468)": enum,
    "maxPcrInterval (p. 468)": integer,
    "audioFramesPerPes (p. 469)": integer,
    "ebpAudioInterval (p. 469)": enum,
    "fragmentTime (p. 469)": number,
    "scte35Pid (p. 469)": integer,
    "pmtInterval (p. 469)": integer,
    "privateMetadataPid (p. 469)": integer,
    "segmentationStyle (p. 470)": enum,
    "audioBufferModel (p. 469)": enum,
    "programNumber (p. 470)": integer,
    "dvbNitSettings (p. 470)": {
        "networkName (p. 411)": "string",
        "networkId (p. 412)": integer,
        "nitInterval (p. 412)": integer
    },
    "timedMetadataPid (p. 470)": integer,
    "scte35Source (p. 470)": enum,
    "pmtPid (p. 470)": integer,
    "bufferModel (p. 470)": enum,
    "ebpPlacement (p. 471)": enum,
    "dvbSdtSettings (p. 471)": {
        "sdtInterval (p. 412)": integer,
        "serviceName (p. 412)": "string",
        "serviceProviderName (p. 412)": "string",
        "outputSdt (p. 412)": enum
    },
    "nullPacketBitrate (p. 471)": number,
    "pcrPid (p. 471)": integer,
    "minEbpInterval (p. 471)": integer,
    "transportStreamId (p. 471)": integer,
    "videoPid (p. 472)": integer,
    "pcrControl (p. 472)": enum,
"esRateInPes (p. 472)": enum,
"segmentationMarkers (p. 472)": enum,
"dvbTdtSettings (p. 472)": {
    "tdtInterval (p. 417)": integer
},
"patInterval (p. 472)": integer,
"dvbSubPids (p. 472)": [ integer
}]
"movSettings (p. 407)": {
    "reference (p. 476)": enum,
    "paddingControl (p. 476)": enum,
    "mpeg2FourCCControl (p. 476)": enum,
    "cslgAtom (p. 476)": enum,
    "clapAtom (p. 476)": enum
},
"f4vSettings (p. 407)": {
    "moovPlacement (p. 423)": enum
}
},
"preset (p. 489)": "string",
"outputSettings (p. 489)": {
    "hlsSettings (p. 491)": {
        "iFrameOnlyManifest (p. 451)": enum,
        "audioRenditionSets (p. 451)": "string",
        "audioTrackType (p. 452)": enum,
        "segmentModifier (p. 452)": "string",
        "audioGroupId (p. 452)": "string"
    }
},
"captionDescriptions (p. 489)": [
    {
        "captionSelectorName (p. 398)": "string",
        "languageDescription (p. 398)": "string",
        "destinationSettings (p. 399)": {
            "burninDestinationSettings (p. 399)": {
                "xPosition (p. 394)": integer,
                "backgroundColor (p. 395)": enum,
                "teletextSpacing (p. 395)": enum,
                "yPosition (p. 395)": integer,
                "backgroundColor (p. 395)": integer,
                "teletextSpacing (p. 395)": enum,
                "fontOpacity (p. 395)": integer,
                "fontResolution (p. 395)": integer,
                "shadowOpacity (p. 395)": integer,
                "shadowOffset (p. 396)": integer,
                "outlineSize (p. 396)": integer,
                "outlineColor (p. 396)": enum,
                "fontSize (p. 396)": integer,
                "shadowXOffset (p. 396)": integer,
                "alignment (p. 397)": enum,
                "shadowColor (p. 397)": enum,
                "foregroundColor (p. 397)": enum
            },
            "teletextDestinationSettings (p. 399)": {
                "pageNumber (p. 498)": "string"
            },
            "ttmlDestinationSettings (p. 399)": {
                "stylePassthrough (p. 501)": enum
            },
            "destinationType (p. 399)": enum,
            "dvbSubDestinationSettings (p. 399)": {
                "xPosition (p. 413)": integer,
                "backgroundColor (p. 413)": enum,
                "teletextSpacing (p. 413)": enum,
                "yPosition (p. 413)": integer,
"backgroundOpacity (p. 413)": integer,
"fontOpacity (p. 413)": integer,
"fontResolution (p. 414)": integer,
"shadowOpacity (p. 414)": integer,
"shadowYOffset (p. 414)": integer,
"outlineSize (p. 414)": integer,
"outlineColor (p. 414)": enum,
"fontSize (p. 414)": integer,
"shadowXOffset (p. 415)": integer,
"alignment (p. 415)": enum,
"shadowColor (p. 415)": enum,
"fontColor (p. 415)": enum
},
"sccDestinationSettings (p. 399)": {
  "framerate (p. 496)": enum
},
"languageCode (p. 399)": enum,
"customLanguageCode (p. 399)": "string"
],
"nameModifier (p. 489)": "string"
},
"outputGroupSettings (p. 490)": {
  "dashIsoGroupSettings (p. 490)": {
    "fragmentLength (p. 409)": integer,
    "baseUrl (p. 409)": "string",
    "minBufferTime (p. 409)": integer,
    "encryption (p. 409)": {
      "spekeKeyProvider (p. 409)": {
        "resourceId (p. 496)": "string",
        "systemIds (p. 497)": ["string"
      },
      "url (p. 497)": "string"
    },
    "destination (p. 409)": "string",
    "segmentLength (p. 410)": integer,
    "segmentControl (p. 410)": enum,
    "hbbtvCompliance (p. 410)": enum
  },
  "fileGroupSettings (p. 490)": {
    "destination (p. 423)": "string"
  },
  "msSmoothGroupSettings (p. 491)": {
    "fragmentLength (p. 485)": integer,
    "encryption (p. 485)": {
      "spekeKeyProvider (p. 485)": {
        "resourceId (p. 496)": "string",
        "systemIds (p. 497)": ["string"
      },
      "url (p. 497)": "string"
    },
    "audioDeduplication (p. 485)": enum,
    "manifestEncoding (p. 486)": enum,
    "destination (p. 486)": "string"
  },
  "cmafGroupSettings (p. 491)": {
    "writeHlsManifest (p. 402)": enum,
    "writeDashManifest (p. 402)": enum,
    "fragmentLength (p. 402)": integer,
    "streamInfResolution (p. 402)": enum,
"minBufferTime (p. 402)" : integer,
"clientCache (p. 403)" : enum,
"codecSpecification (p. 403)" : enum,
"destination (p. 403)" : "string",
"segmentControl (p. 403)" : enum,
"baseUrl (p. 403)" : "string",
"encryption (p. 403)" : {
    "initializationVectorInManifest (p. 401)" : enum,
    "staticKeyProvider (p. 401)" : {
        "keyFormatVersions (p. 497)" : "string",
        "keyFormat (p. 497)" : "string",
        "staticKeyValue (p. 497)" : "string",
        "url (p. 497)" : "string"
    },
    "type (p. 402)" : enum,
    "encryptionMethod (p. 402)" : enum
},
"manifestCompression (p. 403)" : enum,
"segmentLength (p. 403)" : integer,
"manifestDurationFormat (p. 404)" : enum
},
"type (p. 491)" : enum,
"hlsGroupSettings (p. 491)" : {
    "segmentsPerSubdirectory (p. 447)" : integer,
    "streamInfResolution (p. 447)" : enum,
    "timestampDeltaMilliseconds (p. 448)" : integer,
    "outputSelection (p. 448)" : enum,
    "captionLanguageMappings (p. 448)" : [ {
        "languageDescription (p. 445)" : "string",
        "captionChannel (p. 445)" : integer,
        "languageCode (p. 445)" : enum,
        "customLanguageCode (p. 446)" : "string"
    } ],
    "clientCache (p. 448)" : enum,
    "codecSpecification (p. 448)" : enum,
    "destination (p. 448)" : "string",
    "timedMetadataId3Frame (p. 448)" : enum,
    "segmentControl (p. 448)" : enum,
    "timedMetadataId3Period (p. 449)" : integer,
    "captionLanguageSetting (p. 449)" : enum,
    "minSegmentLength (p. 449)" : integer,
    "directoryStructure (p. 449)" : enum,
    "programDateTime (p. 449)" : enum,
    "baseUrl (p. 449)" : "string",
    "encryption (p. 449)" : {
        "initializationVectorInManifest (p. 446)" : enum,
        "staticKeyProvider (p. 446)" : {
            "keyFormatVersions (p. 497)" : "string",
            "keyFormat (p. 497)" : "string",
            "staticKeyValue (p. 497)" : "string",
            "url (p. 497)" : "string"
        },
        "type (p. 447)" : enum,
        "encryptionMethod (p. 447)" : enum,
        "spekeKeyProvider (p. 447)" : {
            "resourceId (p. 496)" : "string",
            "systemIds (p. 497)" : [ "string"
        ],
        "url (p. 497)" : "string"
    }
}
"adMarkers (p. 449)": [
  enum
],
"programDateTimePeriod (p. 450)": integer,
"manifestCompression (p. 450)": enum,
"segmentLength (p. 450)": integer,
"manifestDurationFormat (p. 450)": enum
],
"name (p. 490)": "string",
"customName (p. 490)": "string"
},
"nielsenConfiguration (p. 460)": {
  "distributorId (p. 486)": "string",
  "breakoutCode (p. 486)": integer
},
"inputs (p. 461)": [
  {
    "audioSelectors (p. 453)": {
    },
    "audioSelectorGroups (p. 453)": {
    },
    "videoSelector (p. 453)": {
      "colorSpace (p. 505)": enum,
      "hdr10Metadata (p. 505)": {
        "redPrimaryY (p. 442)": integer,
        "greenPrimaryY (p. 443)": integer,
        "whitePointX (p. 443)": integer,
        "maxLuminance (p. 443)": integer,
        "greenPrimaryX (p. 443)": integer,
        "whitePointY (p. 443)": integer,
        "redPrimaryX (p. 444)": integer,
        "bluePrimaryX (p. 444)": integer,
        "bluePrimaryY (p. 444)": integer,
        "maxFrameAverageLightLevel (p. 444)": integer,
        "maxContentLightLevel (p. 444)": integer,
        "minLuminance (p. 445)": integer
      },
      "programNumber (p. 505)": integer,
      "pid (p. 505)": integer,
      "colorSpaceUsage (p. 506)": enum
    },
    "denoiseFilter (p. 453)": enum,
    "filterEnable (p. 453)": enum,
    "deblockFilter (p. 453)": enum,
    "filterStrength (p. 453)": integer,
    "programNumber (p. 454)": integer,
    "timecodeSource (p. 454)": enum,
    "captionSelectors (p. 454)": {
    },
    "fileInput (p. 454)": "string",
    "inputClippings (p. 454)": [
      {
        "startTimecode (p. 455)": "string",
        "endTimecode (p. 455)": "string"
      }
    ],
    "psiControl (p. 455)": enum
  },
  "timedMetadataInsertion (p. 461)": {
    "id3Insertions (p. 500)": [
      {
        "id3 (p. 452)": "string",
        "timecode (p. 452)": "string"
      }
    ]
  }
]
Example ExceptionBody

```json
{
  "message": "string"
}
```

Properties

**AacAudioDescriptionBroadcasterMix (enum)**

Choose BROADCASTER_MIXED_AD when the input contains pre-mixed main audio + audio description (AD) as a stereo pair. The value for AudioType will be set to 3, which signals to downstream systems that this stream contains "broadcaster mixed AD". Note that the input received by the encoder must contain pre-mixed audio; the encoder does not perform the mixing. When you choose BROADCASTER_MIXED_AD, the encoder ignores any values you provide in AudioType and FollowInputAudioType. Choose NORMAL when the input does not contain pre-mixed audio + audio description (AD). In this case, the encoder will use any values you provide for AudioType and FollowInputAudioType.

- BROADCASTER_MIXED_AD
- NORMAL
**AacCodecProfile (enum)**

AAC Profile.

- LC
- HEV1
- HEV2

**AacCodingMode (enum)**

Mono (Audio Description), Mono, Stereo, or 5.1 channel layout. Valid values depend on rate control mode and profile. "1.0 - Audio Description (Receiver Mix)" setting receives a stereo description plus control track and emits a mono AAC encode of the description track, with control data emitted in the PES header as per ETSI TS 101 154 Annex E.

- AD_RECEIVER_MIX
- CODING_MODE_1_0
- CODING_MODE_1_1
- CODING_MODE_2_0
- CODING_MODE_5_1

**AacRateControlMode (enum)**

Rate Control Mode.

- CBR
- VBR

**AacRawFormat (enum)**

Enables LATM/LOAS AAC output. Note that if you use LATM/LOAS AAC in an output, you must choose "No container" for the output container.

- LATM_LOAS
- NONE

**AacSettings**

**vbrQuality**

- **Type:** string
  - **Required:** False

**codecProfile**

- **Type:** string
  - **Required:** False

**codingMode**

- **Type:** string
  - **Required:** True
specification
  Type: string
  Required: False

bitrate
Average bitrate in bits/second. Defaults and valid values depend on rate control mode and profile.
  Type: integer
  Required: False
  Minimum: 6000
  Maximum: 1024000

rawFormat
  Type: string
  Required: False

rateControlMode
  Type: string
  Required: False

sampleRate
Sample rate in Hz. Valid values depend on rate control mode and profile.
  Type: integer
  Required: True
  Minimum: 8000
  Maximum: 96000

audioDescriptionBroadcasterMix
  Type: string
  Required: False

AacSpecification (enum)
Use MPEG-2 AAC instead of MPEG-4 AAC audio for raw or MPEG-2 Transport Stream containers.
  MPEG2
  MPEG4

AacVbrQuality (enum)
VBR Quality Level - Only used if rate_control_mode is VBR.
  LOW
  MEDIUM_LOW
  MEDIUM_HIGH
  HIGH
**Ac3BitstreamMode (enum)**

Specifies the "Bitstream Mode" (bsmod) for the emitted AC-3 stream. See ATSC A/52-2012 for background on these values.

- COMPLETE_MAIN
- COMMENTARY
- DIALOGUE
- EMERGENCY
- HEARING_IMPAIRED
- MUSIC_AND_EFFECTS
- VISUALLY_IMPAIRED
- VOICE_OVER

**Ac3CodingMode (enum)**

Dolby Digital coding mode. Determines number of channels.

- CODING_MODE_1_0
- CODING_MODE_1_1
- CODING_MODE_2_0
- CODING_MODE_3_2_LFE

**Ac3DynamicRangeCompressionProfile (enum)**

If set to FILM_STANDARD, adds dynamic range compression signaling to the output bitstream as defined in the Dolby Digital specification.

- FILM_STANDARD
- NONE

**Ac3LfeFilter (enum)**

Applies a 120Hz lowpass filter to the LFE channel prior to encoding. Only valid with 3_2_LFE coding mode.

- ENABLED
- DISABLED

**Ac3MetadataControl (enum)**

When set to FOLLOW_INPUT, encoder metadata will be sourced from the DD, DD+, or DolbyE decoder that supplied this audio data. If audio was not supplied from one of these streams, then the static metadata settings will be used.

- FOLLOW_INPUT
- USE_CONFIGURED

**Ac3Settings**

*dynamicRangeCompressionProfile*

Type: string
Required: False
**dialnorm**

Sets the dialnorm for the output. If blank and input audio is Dolby Digital, dialnorm will be passed through.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 31

**codingMode**

- **Type:** string
- **Required:** False

**metadataControl**

- **Type:** string
- **Required:** False

**bitrate**

Average bitrate in bits/second. Valid bitrates depend on the coding mode.

- **Type:** integer
- **Required:** False
- **Minimum:** 64000
- **Maximum:** 640000

**lfeFilter**

- **Type:** string
- **Required:** False

**bitstreamMode**

- **Type:** string
- **Required:** False

**sampleRate**

Sample rate in hz. Sample rate is always 48000.

- **Type:** integer
- **Required:** False
- **Minimum:** 48000
- **Maximum:** 48000

**AfdSignaling (enum)**

This setting only applies to H.264 and MPEG2 outputs. Use Insert AFD signaling (AfdSignaling) to specify whether the service includes AFD values in the output video data and what those values are. * Choose None to remove all AFD values from this output. * Choose Fixed to ignore input AFD values and instead...
encode the value specified in the job. * Choose Auto to calculate output AFD values based on the input
AFD scaler data.

   NONE
   AUTO
   FIXED

**AiffSettings**

**channels**

Set Channels to specify the number of channels in this output audio track. Choosing Mono in the console
will give you 1 output channel; choosing Stereo will give you 2. In the API, valid values are 1 and 2.

   Type: integer
   Required: False
   Minimum: 1
   Maximum: 2

**bitDepth**

Specify Bit depth (BitDepth), in bits per sample, to choose the encoding quality for this audio track.

   Type: integer
   Required: False
   Minimum: 16
   Maximum: 24

**sampleRate**

Sample rate in hz.

   Type: integer
   Required: False
   Minimum: 8000
   Maximum: 192000

**AncillarySourceSettings**

**sourceAncillaryChannelNumber**

Specifies the 608 channel number in the ancillary data track from which to extract captions. Unused for passthrough.

   Type: integer
   Required: False
   Minimum: 1
   Maximum: 4

**AntiAlias (enum)**

Enable Anti-alias (AntiAlias) to enhance sharp edges in video output when your input resolution is much
larger than your output resolution. Default is enabled.
DISABLED
ENABLED

**AudioCodec (enum)**

Type of Audio codec.

AAC
MP2
WAV
AIFF
AC3
EAC3
PASSTHROUGH

**AudioCodecSettings**

codec

Type: string
Required: True

**wavSettings**

Type: WavSettings (p. 506)
Required: False

**aacSettings**

Type: AacSettings (p. 385)
Required: False

**ac3Settings**

Type: Ac3Settings (p. 387)
Required: False

**aiffSettings**

Type: AiffSettings (p. 389)
Required: False

**eac3Settings**

Type: Eac3Settings (p. 419)
Required: False

**mp2Settings**

Type: Mp2Settings (p. 477)
**Required**: False

**AudioDefaultSelection (enum)**

Enable this setting on one audio selector to set it as the default for the job. The service uses this default for outputs where it can't find the specified input audio. If you don't set a default, those outputs have no audio.

- DEFAULT
- NOT_DEFAULT

**AudioDescription**

**audioTypeControl**

- **Type**: string
- **Required**: False

**languageCodeControl**

- **Type**: string
- **Required**: False

**remixSettings**

Advanced audio remixing settings.

- **Type**: RemixSettings (p. 495)
- **Required**: False

**audioType**

Applies only if Follow Input Audio Type is unchecked (false). A number between 0 and 255. The following are defined in ISO-IEC 13818-1: 0 = Undefined, 1 = Clean Effects, 2 = Hearing Impaired, 3 = Visually Impaired Commentary, 4-255 = Reserved.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 255

**audioSourceName**

Specifies which audio data to use from each input. In the simplest case, specify an "Audio Selector":#inputs-audio_selector by name based on its order within each input. For example if you specify "Audio Selector 3", then the third audio selector will be used from each input. If an input does not have an "Audio Selector 3", then the audio selector marked as "default" in that input will be used. If there is no audio selector marked as "default", silence will be inserted for the duration of that input. Alternatively, an "Audio Selector Group":#inputs-audio_selector_group name may be specified, with similar default/silence behavior. If no audio_source_name is specified, then "Audio Selector 1" will be chosen automatically.

- **Type**: string
**Properties**

**codecSettings**
- **Type:** AudioCodecSettings (p. 390)
- **Required:** True

**languageCode**
Indicates the language of the audio output track. The ISO 639 language specified in the 'Language Code' drop down will be used when 'Follow Input Language Code' is not selected or when 'Follow Input Language Code' is selected but there is no ISO 639 language code specified by the input.
- **Type:** string
- **Required:** False
- **Pattern:** ^\[\w\s\]*$

**streamName**
Used for MS Smooth and Apple HLS outputs. Indicates the name displayed by the player (eg. English, or Director Commentary). Alphanumeric characters, spaces, and underscore are legal.
- **Type:** string
- **Required:** False
- **Pattern:** ^[^\w\s]\*$

**audioNormalizationSettings**
- **Type:** AudioNormalizationSettings (p. 393)
- **Required:** False

**customLanguageCode**
Specify the language for this audio output track, using the ISO 639-2 or ISO 639-3 three-letter language code. The language specified will be used when 'Follow Input Language Code' is not selected or when 'Follow Input Language Code' is selected but there is no ISO 639 language code specified by the input.
- **Type:** string
- **Required:** False
- **Pattern:** ^[A-Za-z]{3}$

**AudioLanguageCodeControl (enum)**
Choosing FOLLOW_INPUT will cause the ISO 639 language code of the output to follow the ISO 639 language code of the input. The language specified for languageCode' will be used when USE_CONFIGURED is selected or when FOLLOW_INPUT is selected but there is no ISO 639 language code specified by the input.
- FOLLOW_INPUT
- USE_CONFIGURED

**AudioNormalizationAlgorithm (enum)**
Audio normalization algorithm to use. 1770-1 conforms to the CALM Act specification, 1770-2 conforms to the EBU R-128 specification.
ITU_BS_1770_1
ITU_BS_1770_2

AudioNormalizationAlgorithmControl (enum)

When enabled the output audio is corrected using the chosen algorithm. If disabled, the audio will be measured but not adjusted.

    CORRECT_AUDIO
    MEASURE_ONLY

AudioNormalizationLoudnessLogging (enum)

If set to LOG, log each output's audio track loudness to a CSV file.

    LOG
    DONT_LOG

AudioNormalizationPeakCalculation (enum)

If set to TRUE_PEAK, calculate and log the TruePeak for each output's audio track loudness.

    TRUE_PEAK
    NONE

AudioNormalizationSettings

targetLkfs

Target LKFS(loudness) to adjust volume to. If no value is entered, a default value will be used according to the chosen algorithm. The CALM Act (1770-1) recommends a target of -24 LKFS. The EBU R-128 specification (1770-2) recommends a target of -23 LKFS.

    Type: number
    Required: False
    Format: float
    Minimum: -59.0
    Maximum: 0.0

algorithmControl

    Type: string
    Required: False

loudnessLogging

    Type: string
    Required: False

peakCalculation

    Type: string
    Required: False
**correctionGateLevel**

Content measuring above this level will be corrected to the target level. Content measuring below this level will not be corrected. Gating only applies when not using real_time_correction.

- **Type**: integer
- **Required**: False
- **Minimum**: -70
- **Maximum**: 0

**algorithm**

- **Type**: string
- **Required**: False

**AudioSelectorType (enum)**

Specifies the type of the audio selector.

- PID
- TRACK
- LANGUAGE_CODE

**AudioTypeControl (enum)**

When set to FOLLOW_INPUT, if the input contains an ISO 639 audio_type, then that value is passed through to the output. If the input contains no ISO 639 audio_type, the value in Audio Type is included in the output. Otherwise the value in Audio Type is included in the output. Note that this field and audioType are both ignored if audioDescriptionBroadcasterMix is set to BROADCASTER_MIXED_AD.

- FOLLOW_INPUT
- USE_CONFIGURED

**AvailBlanking**

**availBlankingImage**

Blanking image to be used. Leave empty for solid black. Only bmp and png images are supported.

- **Type**: string
- **Required**: False
- **Pattern**: ^s3://.*/\(.\.*\)\.(bmp|BMP|png|PNG)$

**BurninDestinationSettings**

**xPosition**

Specifies the horizontal position of the caption relative to the left side of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the left of the output. If no explicit x_position is provided, the horizontal caption position will be determined by the alignment parameter. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
**Properties**

**backgroundColor**

- **Type:** string
- **Required:** False

**teletextSpacing**

- **Type:** string
- **Required:** False

**yPosition**

Specifies the vertical position of the caption relative to the top of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the top of the output. If no explicit y_position is provided, the caption will be positioned towards the bottom of the output. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
- **Maximum:** 2147483647

**backgroundOpacity**

Specifies the opacity of the background rectangle. 255 is opaque; 0 is transparent. Leaving this parameter blank is equivalent to setting it to 0 (transparent). All burn-in and DVB-Sub font settings must match.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
- **Maximum:** 255

**fontOpacity**

Specifies the opacity of the burned-in captions. 255 is opaque; 0 is transparent. All burn-in and DVB-Sub font settings must match.

- **Type:** integer
- **Required:** True
- **Minimum:** 0
- **Maximum:** 255

**fontResolution**

Font resolution in DPI (dots per inch); default is 96 dpi. All burn-in and DVB-Sub font settings must match.

- **Type:** integer
Properties

**Required**: False

**Minimum**: 96

**Maximum**: 600

**shadowOpacity**

Specifies the opacity of the shadow. 255 is opaque; 0 is transparent. Leaving this parameter blank is equivalent to setting it to 0 (transparent). All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 255

**shadowYOffset**

Specifies the vertical offset of the shadow relative to the captions in pixels. A value of -2 would result in a shadow offset 2 pixels above the text. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: False
- **Minimum**: -2147483648
- **Maximum**: 2147483647

**outlineSize**

Specifies font outline size in pixels. This option is not valid for source captions that are either 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: True
- **Minimum**: 0
- **Maximum**: 10

**outlineColor**

- **Type**: string
- **Required**: True

**fontSize**

A positive integer indicates the exact font size in points. Set to 0 for automatic font size selection. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 96

**shadowXOffset**

Specifies the horizontal offset of the shadow relative to the captions in pixels. A value of -2 would result in a shadow offset 2 pixels to the left. All burn-in and DVB-Sub font settings must match.
**Type**: integer
**Required**: False
**Minimum**: -2147483648
**Maximum**: 2147483647

**alignment**

**Type**: string
**Required**: True

**shadowColor**

**Type**: string
**Required**: False

**fontColor**

**Type**: string
**Required**: False

**BurninSubtitleAlignment (enum)**

If no explicit x_position or y_position is provided, setting alignment to centered will place the captions at the bottom center of the output. Similarly, setting a left alignment will align captions to the bottom left of the output. If x and y positions are given in conjunction with the alignment parameter, the font will be justified (either left or centered) relative to those coordinates. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- CENTERED
- LEFT

**BurninSubtitleBackgroundColor (enum)**

Specifies the color of the rectangle behind the captions. All burn-in and DVB-Sub font settings must match.

- NONE
- BLACK
- WHITE

**BurninSubtitleFontColor (enum)**

Specifies the color of the burned-in captions. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- WHITE
- BLACK
- YELLOW
- RED
- GREEN
BLUE

**BurninSubtitleOutlineColor (enum)**

Specifies font outline color. This option is not valid for source captions that are either 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

BLACK
WHITE
YELLOW
RED
GREEN
BLUE

**BurninSubtitleShadowColor (enum)**

Specifies the color of the shadow cast by the captions. All burn-in and DVB-Sub font settings must match.

NONE
BLACK
WHITE

**BurninSubtitleTeletextSpacing (enum)**

Only applies to jobs with input captions in Teletext or STL formats. Specify whether the spacing between letters in your captions is set by the captions grid or varies depending on letter width. Choose fixed grid to conform to the spacing specified in the captions file more accurately. Choose proportional to make the text easier to read if the captions are closed caption.

FIXED_GRID
PROPORTIONAL

**CaptionDescription**

**captionSelectorName**

Specifies which "Caption Selector" to use from each input when generating captions. The name should be of the format "Caption Selector <N>", which denotes that the Nth Caption Selector will be used from each input.

- **Type:** string
- **Required:** True

**languageDescription**

Human readable information to indicate captions available for players (eg. English, or Spanish). Alphanumeric characters, spaces, and underscore are legal.

- **Type:** string
- **Required:** False
- **Pattern:** ^[^\s_]*$
**destinationSettings**
- **Type:** CaptionDestinationSettings (p. 399)
- **Required:** True

**languageCode**
Indicates the language of the caption output track.
- **Type:** string
- **Required:** False

**customLanguageCode**
Indicates the language of the caption output track, using the ISO 639-2 or ISO 639-3 three-letter language code.
- **Type:** string
- **Required:** False
- **Pattern:** ^[A-Za-z]{3}$

**CaptionDestinationSettings**

**burninDestinationSettings**
- **Type:** BurninDestinationSettings (p. 394)
- **Required:** False

**teletextDestinationSettings**
- **Type:** TeletextDestinationSettings (p. 498)
- **Required:** False

**ttmIDestinationSettings**
- **Type:** TtmlDestinationSettings (p. 501)
- **Required:** False

**destinationType**
- **Type:** string
- **Required:** True

**dvbSubDestinationSettings**
- **Type:** DvbSubDestinationSettings (p. 413)
- **Required:** False

**sccDestinationSettings**
- **Type:** SccDestinationSettings (p. 496)
- **Required:** False
CaptionDestinationType (enum)

Type of Caption output, including Burn-In, Embedded, SCC, SRT, TTML, WebVTT, DVB-Sub, Teletext.

- BURN_IN
- DVB_SUB
- EMBEDDED
- SCC
- SRT
- TELETEXT
- TTML
- WEBVTT

CaptionSourceSettings

fileSourceSettings

Type: FileSourceSettings (p. 423)
Required: False

ancillarySourceSettings

Type: AncillarySourceSettings (p. 389)
Required: False

embeddedSourceSettings

Type: EmbeddedSourceSettings (p. 422)
Required: False

sourceType

Type: string
Required: True

dvbSubSourceSettings

Type: DvbSubSourceSettings (p. 415)
Required: False

teletextSourceSettings

Type: TeletextSourceSettings (p. 498)
Required: False

CaptionSourceType (enum)

Use Source (SourceType) to identify the format of your input captions. The service cannot auto-detect caption format.

- ANCILLARY
- DVB_SUB
ChannelMapping

outputChannels

List of output channels

Type: Array of type OutputChannelMapping (p. 489)
Required: True

CmafClientCache (enum)

When set to ENABLED, sets #EXT-X-ALLOW-CACHE:no tag, which prevents client from saving media segments for later replay.

DISABLED
ENABLED

CmafCodecSpecification (enum)

Specification to use (RFC-6381 or the default RFC-4281) during m3u8 playlist generation.

RFC_6381
RFC_4281

CmafEncryptionSettings

initializationVectorInManifest

Type: string
Required: False

constantInitializationVector

This is a 128-bit, 16-byte hex value represented by a 32-character text string. If this parameter is not set then the Initialization Vector will follow the segment number by default.

Type: string
Required: False
Pattern: ^[0-9a-fA-F]{32}$

staticKeyProvider

Type: StaticKeyProvider (p. 497)
Required: False
### type

- **Type**: string
- **Required**: True

### encryptionMethod

- **Type**: string
- **Required**: False

#### CmafEncryptionType (enum)

Encrypts the segments with the given encryption scheme. Leave blank to disable. Selecting 'Disabled' in the web interface also disables encryption.

**SAMPLE_AES**

### CmafGroupSettings

#### writeHlsManifest

- **Type**: string
- **Required**: False

#### writeDashManifest

- **Type**: string
- **Required**: False

### fragmentLength

Length of fragments to generate (in seconds). Fragment length must be compatible with GOP size and Framerate. Note that fragments will end on the next keyframe after this number of seconds, so actual fragment length may be longer. When Emit Single File is checked, the fragmentation is internal to a single output file and it does not cause the creation of many output files as in other output types.

- **Type**: integer
- **Required**: True
- **Minimum**: 1
- **Maximum**: 2147483647

### streamInfResolution

- **Type**: string
- **Required**: False

### minBufferTime

Minimum time of initially buffered media that is needed to ensure smooth playout.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
Maximum: 2147483647

**clientCache**

- **Type:** string
- **Required:** False

**codecSpecification**

- **Type:** string
- **Required:** False

**destination**

Use **Destination** (Destination) to specify the S3 output location and the output filename base. Destination accepts format identifiers. If you do not specify the base filename in the URI, the service will use the filename of the input file. If your job has multiple inputs, the service uses the filename of the first input file.

- **Type:** string
- **Required:** False
- **Pattern:** ^s3:/\/

**segmentControl**

- **Type:** string
- **Required:** False

**baseUrl**

A partial URI prefix that will be put in the manifest file at the top level **BaseUrl** element. Can be used if streams are delivered from a different URL than the manifest file.

- **Type:** string
- **Required:** False

**encryption**

DRM settings.

- **Type:** CmafEncryptionSettings (p. 401)
- **Required:** False

**manifestCompression**

- **Type:** string
- **Required:** False

**segmentLength**

Use this setting to specify the length, in seconds, of each individual CMAF segment. This value applies to the whole package; that is, to every output in the output group. Note that segments end on the first keyframe after this number of seconds, so the actual segment length might be slightly longer. If you
set Segment control (CmafSegmentControl) to single file, the service puts the content of each output in a single file that has metadata that marks these segments. If you set it to segmented files, the service creates multiple files for each output, each with the content of one segment.

**Type**: integer  
**Required**: True  
**Minimum**: 1  
**Maximum**: 2147483647

**manifestDurationFormat**

**Type**: string  
**Required**: False

**CmafInitializationVectorInManifest (enum)**

The Initialization Vector is a 128-bit number used in conjunction with the key for encrypting blocks. If set to INCLUDE, Initialization Vector is listed in the manifest. Otherwise Initialization Vector is not in the manifest.

**INCLUDE**  
**EXCLUDE**

**CmafKeyProviderType (enum)**

Indicates which type of key provider is used for encryption.

**STATIC_KEY**

**CmafManifestCompression (enum)**

When set to GZIP, compresses HLS playlist.

**GZIP**  
**NONE**

**CmafManifestDurationFormat (enum)**

Indicates whether the output manifest should use floating point values for segment duration.

**FLOATING_POINT**  
**INTEGER**

**CmafSegmentControl (enum)**

When set to SINGLE_FILE, a single output file is generated, which is internally segmented using the Fragment Length and Segment Length. When set to SEGMENTED_FILES, separate segment files will be created.

**SINGLE_FILE**  
**SEGMENTED_FILES**

**CmafStreamInfResolution (enum)**

Include or exclude RESOLUTION attribute for video in EXT-X-STREAM-INF tag of variant manifest.
INCLUDE
EXCLUDE

CmafWriteDASHManifest (enum)
When set to ENABLED, a DASH MPD manifest will be generated for this output.
   DISABLED
   ENABLED

CmafWriteHLSManifest (enum)
When set to ENABLED, an Apple HLS manifest will be generated for this output.
   DISABLED
   ENABLED

ColorCorrector

saturation
Saturation level.
   Type: integer
   Required: False
   Minimum: 1
   Maximum: 100

brightness
Brightness level.
   Type: integer
   Required: False
   Minimum: 1
   Maximum: 100

hdr10Metadata
   Type: Hdr10Metadata (p. 442)
   Required: False

contrast
Contrast level.
   Type: integer
   Required: False
   Minimum: 1
   Maximum: 100

hue
Hue in degrees.
Properties

**Type**: integer  
**Required**: False  
**Minimum**: -180  
**Maximum**: 180

**colorSpaceConversion**

**Type**: string  
**Required**: False

**ColorMetadata (enum)**

Enable insert color metadata (ColorMetadata) to include color metadata in this output. This setting is enabled by default.

- **IGNORE**
- **INSERT**

**ColorSpace (enum)**

If your input video has accurate color space metadata, or if you don’t know about color space, leave this set to the default value FOLLOW. The service will automatically detect your input color space. If your input video has metadata indicating the wrong color space, or if your input video is missing color space metadata that should be there, specify the accurate color space here. If you choose HDR10, you can also correct inaccurate color space coefficients, using the HDR master display information controls. You must also set Color space usage (ColorSpaceUsage) to FORCE for the service to use these values.

- **FOLLOW**
- **REC_601**
- **REC_709**
- **HDR10**
- **HLG_2020**

**ColorSpaceConversion (enum)**

Determines if colorspace conversion will be performed. If set to _None_, no conversion will be performed. If _Force 601_ or _Force 709_ are selected, conversion will be performed for inputs with differing colorspace. An input’s colorspace can be specified explicitly in the "Video Selector".#inputs-video_selector if necessary.

- **NONE**
- **FORCE_601**
- **FORCE_709**
- **FORCE_HDR10**
- **FORCE_HLG_2020**

**ColorSpaceUsage (enum)**

There are two sources for color metadata, the input file and the job configuration (in the Color space and HDR master display information settings). The Color space usage setting controls which takes precedence. FORCE: The system will use color metadata supplied by user, if any. If the user does not supply color metadata, the system will use data from the source. FALLBACK: The system will use color metadata from the source. If source has no color metadata, the system will use user-supplied color metadata values if available.
FORCE
FALLBACK

**ContainerSettings**

container

Type: string
Required: True

mp4Settings

Type: Mp4Settings (p. 478)
Required: False

m3u8Settings

Type: M3u8Settings (p. 473)
Required: False

m2tsSettings

Type: M2tsSettings (p. 468)
Required: False

movSettings

Type: MovSettings (p. 476)
Required: False

f4vSettings

Type: F4vSettings (p. 423)
Required: False

**ContainerType (enum)**

Container for this output. Some containers require a container settings object. If not specified, the default object will be created.

F4V
ISMV
M2TS
M3U8
CMFC
MOV
MP4
MPD
MXF
RAW
CreateJobRequest

settings

Type: JobSettings (p. 460)
Required: True

role

Required. The IAM role you use for creating this job. For details about permissions, see the User Guide topic at the User Guide at http://docs.aws.amazon.com/mediaconvert/latest/ug/iam-role.html.

Type: string
Required: True

clientRequestToken

Idempotency token for CreateJob operation.

Type: string
Required: False

jobTemplate

When you create a job, you can either specify a job template or specify the transcoding settings individually

Type: string
Required: False

userMetadata

User-defined metadata that you want to associate with an MediaConvert job. You specify metadata in key/value pairs.

Type: object
Required: False

queue

Optional. When you create a job, you can specify a queue to send it to. If you don't specify, the job will go to the default queue. For more about queues, see the User Guide topic at http://docs.aws.amazon.com/mediaconvert/latest/ug/what-is.html.

Type: string
Required: False

CreateJobResponse

job

Type: Job (p. 458)
**Required**: False

**DashIsoEncryptionSettings**

**spekeKeyProvider**

*Type*: SpekeKeyProvider (p. 496)

*Required*: True

**DashIsoGroupSettings**

**fragmentLength**

Length of fragments to generate (in seconds). Fragment length must be compatible with GOP size and Framerate. Note that fragments will end on the next keyframe after this number of seconds, so actual fragment length may be longer. When Emit Single File is checked, the fragmentation is internal to a single output file and it does not cause the creation of many output files as in other output types.

*Type*: integer

*Required*: True

*Minimum*: 1

*Maximum*: 2147483647

**baseUrl**

A partial URI prefix that will be put in the manifest (.mpd) file at the top level BaseURL element. Can be used if streams are delivered from a different URL than the manifest file.

*Type*: string

*Required*: False

**minBufferTime**

Minimum time of initially buffered media that is needed to ensure smooth playout.

*Type*: integer

*Required*: False

*Minimum*: 0

*Maximum*: 2147483647

**encryption**

DRM settings.

*Type*: DashIsoEncryptionSettings (p. 409)

*Required*: False

**destination**

Use Destination (Destination) to specify the S3 output location and the output filename base. Destination accepts format identifiers. If you do not specify the base filename in the URI, the service will use the filename of the input file. If your job has multiple inputs, the service uses the filename of the first input file.
Properties

segmentLength

Length of mpd segments to create (in seconds). Note that segments will end on the next keyframe after this number of seconds, so actual segment length may be longer. When Emit Single File is checked, the segmentation is internal to a single output file and it does not cause the creation of many output files as in other output types.

Type: integer
Required: True
Minimum: 1
Maximum: 2147483647

segmentControl

Type: string
Required: False

hbbtvCompliance

Type: string
Required: False

DashIsoHbbtvCompliance (enum)

Supports HbbTV specification as indicated

HBBTV_1_5
NONE

DashIsoSegmentControl (enum)

When set to SINGLE_FILE, a single output file is generated, which is internally segmented using the Fragment Length and Segment Length. When set to SEGMENTED_FILES, separate segment files will be created.

SINGLE_FILE
SEGMENTED_FILES

DeinterlaceAlgorithm (enum)

Only applies when you set Deinterlace (DeinterlaceMode) to Deinterlace (DEINTERLACE) or Adaptive (ADAPTIVE). Motion adaptive interpolate (INTERPOLATE) produces sharper pictures, while blend (BLEND) produces smoother motion. Use (INTERPOLATE_TICKER) OR (BLEND_TICKER) if your source file includes a ticker, such as a scrolling headline at the bottom of the frame.

INTERPOLATE
INTERPOLATE_TICKER
BLEND
BLEND_TICKER
Deinterlacer

mode

Type: string
Required: False

control

Type: string
Required: False

algorithm

Type: string
Required: False

DeinterlacerControl (enum)

- When set to NORMAL (default), the deinterlacer does not convert frames that are tagged in metadata as progressive. It will only convert those that are tagged as some other type. - When set to FORCE_ALL_FRAMES, the deinterlacer converts every frame to progressive - even those that are already tagged as progressive. Turn Force mode on only if there is a good chance that the metadata has tagged frames as progressive when they are not progressive. Do not turn on otherwise; processing frames that are already progressive into progressive will probably result in lower quality video.

FORCE_ALL_FRAMES
NORMAL

DeinterlacerMode (enum)

Use Deinterlacer (DeinterlaceMode) to choose how the service will do deinterlacing. Default is Deinterlace. - Deinterlace converts interlaced to progressive. - Inverse telecine converts Hard Telecine 29.97i to progressive 23.976p. - Adaptive auto-detects and converts to progressive.

DEINTERLACE
INVERSE_TELECINE
ADAPTIVE

DropFrameTimecode (enum)

Applies only to 29.97 fps outputs. When this feature is enabled, the service will use drop-frame timecode on outputs. If it is not possible to use drop-frame timecode, the system will fall back to non-drop-frame. This setting is enabled by default when Timecode insertion (TimecodeInsertion) is enabled.

DISABLED
ENABLED

DvbNitSettings

networkName

The network name text placed in the network_name_descriptor inside the Network Information Table. Maximum length is 256 characters.
networkId

The numeric value placed in the Network Information Table (NIT).

Type: integer
Required: True
Minimum: 0
Maximum: 65535

nitInterval

The number of milliseconds between instances of this table in the output transport stream.

Type: integer
Required: True
Minimum: 25
Maximum: 10000

DvbSdtSettings

sdInterval

The number of milliseconds between instances of this table in the output transport stream.

Type: integer
Required: False
Minimum: 25
Maximum: 2000

serviceName

The service name placed in the service_descriptor in the Service Description Table. Maximum length is 256 characters.

Type: string
Required: False

serviceProviderName

The service provider name placed in the service_descriptor in the Service Description Table. Maximum length is 256 characters.

Type: string
Required: False

outputSdt

Type: string
Required: False
**DvbSubDestinationSettings**

**xPosition**

Specifies the horizontal position of the caption relative to the left side of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the left of the output. If no explicit x_position is provided, the horizontal caption position will be determined by the alignment parameter. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
- **Maximum:** 2147483647

**backgroundColor**

- **Type:** string
- **Required:** False

**teletextSpacing**

- **Type:** string
- **Required:** False

**yPosition**

Specifies the vertical position of the caption relative to the top of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the top of the output. If no explicit y_position is provided, the caption will be positioned towards the bottom of the output. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
- **Maximum:** 2147483647

**backgroundOpacity**

Specifies the opacity of the background rectangle. 255 is opaque; 0 is transparent. Leaving this parameter blank is equivalent to setting it to 0 (transparent). All burn-in and DVB-Sub font settings must match.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
- **Maximum:** 255

**fontOpacity**

Specifies the opacity of the burned-in captions. 255 is opaque; 0 is transparent. All burn-in and DVB-Sub font settings must match.

- **Type:** integer
Properties

**Required**: True
**Minimum**: 0
**Maximum**: 255

### fontResolution

Font resolution in DPI (dots per inch); default is 96 dpi. All burn-in and DVB-Sub font settings must match.

**Type**: integer
**Required**: False
**Minimum**: 96
**Maximum**: 600

### shadowOpacity

Specifies the opacity of the shadow. 255 is opaque; 0 is transparent. Leaving this parameter blank is equivalent to setting it to 0 (transparent). All burn-in and DVB-Sub font settings must match.

**Type**: integer
**Required**: False
**Minimum**: 0
**Maximum**: 255

### shadowYOffset

Specifies the vertical offset of the shadow relative to the captions in pixels. A value of -2 would result in a shadow offset 2 pixels above the text. All burn-in and DVB-Sub font settings must match.

**Type**: integer
**Required**: False
**Minimum**: -2147483648
**Maximum**: 2147483647

### outlineSize

Specifies font outline size in pixels. This option is not valid for source captions that are either 608/ embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

**Type**: integer
**Required**: True
**Minimum**: 0
**Maximum**: 10

### outlineColor

**Type**: string
**Required**: True

### fontSize

A positive integer indicates the exact font size in points. Set to 0 for automatic font size selection. All burn-in and DVB-Sub font settings must match.
Properties

Type: integer
Required: False
Minimum: 0
Maximum: 96

shadowXOffset

Specifies the horizontal offset of the shadow relative to the captions in pixels. A value of -2 would result in a shadow offset 2 pixels to the left. All burn-in and DVB-Sub font settings must match.

Type: integer
Required: False
Minimum: -2147483648
Maximum: 2147483647

alignment

Type: string
Required: True

shadowColor

Type: string
Required: False

fontColor

Type: string
Required: False

DvbSubSourceSettings

pid

When using DVB-Sub with Burn-In or SMPTE-TT, use this PID for the source content. Unused for DVB-Sub passthrough. All DVB-Sub content is passed through, regardless of selectors.

Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

DvbSubtitleAlignment (enum)

If no explicit x_position or y_position is provided, setting alignment to centered will place the captions at the bottom center of the output. Similarly, setting a left alignment will align captions to the bottom left of the output. If x and y positions are given in conjunction with the alignment parameter, the font will be justified (either left or centered) relative to those coordinates. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

CENTERED
**Properties**

**DvbSubtitleBackgroundColor (enum)**

Specifies the color of the rectangle behind the captions. All burn-in and DVB-Sub font settings must match.

- NONE
- BLACK
- WHITE

**DvbSubtitleFontColor (enum)**

Specifies the color of the burned-in captions. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- WHITE
- BLACK
- YELLOW
- RED
- GREEN
- BLUE

**DvbSubtitleOutlineColor (enum)**

Specifies font outline color. This option is not valid for source captions that are either 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- BLACK
- WHITE
- YELLOW
- RED
- GREEN
- BLUE

**DvbSubtitleShadowColor (enum)**

Specifies the color of the shadow cast by the captions. All burn-in and DVB-Sub font settings must match.

- NONE
- BLACK
- WHITE

**DvbSubtitleTeletextSpacing (enum)**

Only applies to jobs with input captions in Teletext or STL formats. Specify whether the spacing between letters in your captions is set by the captions grid or varies depending on letter width. Choose fixed grid to conform to the spacing specified in the captions file more accurately. Choose proportional to make the text easier to read if the captions are closed caption.

- FIXED_GRID
PROPORTIONAL

DvbTdtSettings

tdtInterval

The number of milliseconds between instances of this table in the output transport stream.

Type: integer
Required: True
Minimum: 1000
Maximum: 30000

Eac3AttenuationControl (enum)

If set to ATTENUATE_3_DB, applies a 3 dB attenuation to the surround channels. Only used for 3/2 coding mode.

ATTENUATE_3_DB
NONE

Eac3BitstreamMode (enum)

Specifies the "Bitstream Mode" (bsmod) for the emitted E-AC-3 stream. See ATSC A/52-2012 (Annex E) for background on these values.

COMPLETE_MAIN
COMMENTARY
EMERGENCY
HEARING_IMPAIRED
VISUALLY_IMPAIRED

Eac3CodingMode (enum)

Dolby Digital Plus coding mode. Determines number of channels.

CODING_MODE_1_0
CODING_MODE_2_0
CODING_MODE_3_2

Eac3DcFilter (enum)

Activates a DC highpass filter for all input channels.

ENABLED
DISABLED

Eac3DynamicRangeCompressionLine (enum)

Enables Dynamic Range Compression that restricts the absolute peak level for a signal.

NONE
FILM_STANDARD
FILM_LIGHT
MUSIC_STANDARD
MUSIC_LIGHT
SPEECH

**Eac3DynamicRangeCompressionRf (enum)**

Enables Heavy Dynamic Range Compression, ensures that the instantaneous signal peaks do not exceed specified levels.

NONE
FILM_STANDARD
FILM_LIGHT
MUSIC_STANDARD
MUSIC_LIGHT
SPEECH

**Eac3LfeControl (enum)**

When encoding 3/2 audio, controls whether the LFE channel is enabled.

LFE
NO_LFE

**Eac3LfeFilter (enum)**

Applies a 120Hz lowpass filter to the LFE channel prior to encoding. Only valid with 3_2_LFE coding mode.

ENABLED
DISABLED

**Eac3MetadataControl (enum)**

When set to FOLLOW_INPUT, encoder metadata will be sourced from the DD, DD+, or DolbyE decoder that supplied this audio data. If audio was not supplied from one of these streams, then the static metadata settings will be used.

FOLLOW_INPUT
USE_CONFIGURED

**Eac3PassthroughControl (enum)**

When set to WHEN_POSSIBLE, input DD+ audio will be passed through if it is present on the input. This detection is dynamic over the life of the transcode. Inputs that alternate between DD+ and non-DD+ content will have a consistent DD+ output as the system alternates between passthrough and encoding.

WHEN_POSSIBLE
NO_PASSTHROUGH

**Eac3PhaseControl (enum)**

Controls the amount of phase-shift applied to the surround channels. Only used for 3/2 coding mode.
SHIFT_90_DEGREES
NO_SHIFT

**Eac3Settings**

**dialnorm**
Sets the dialnorm for the output. If blank and input audio is Dolby Digital Plus, dialnorm will be passed through.

Type: integer  
Required: False  
Minimum: 1  
Maximum: 31

**passthroughControl**

Type: string  
Required: False

**metadataControl**

Type: string  
Required: False

**bitrate**
Average bitrate in bits/second. Valid bitrates depend on the coding mode.

Type: integer  
Required: False  
Minimum: 64000  
Maximum: 640000

**dynamicRangeCompressionRf**

Type: string  
Required: False

**sampleRate**
Sample rate in hz. Sample rate is always 48000.

Type: integer  
Required: False  
Minimum: 48000  
Maximum: 48000

**surroundExMode**

Type: string  
Required: False
**Properties**

- **ltRtSurroundMixLevel**
  - Left total/Right total surround mix level. Only used for 3/2 coding mode. Valid values: -1.5 -3.0 -4.5 -6.0 -60
  - Type: number
  - Required: False
  - Format: float
  - Minimum: -60.0
  - Maximum: -1.5

- **dynamicRangeCompressionLine**
  - Type: string
  - Required: False

- **lfeControl**
  - Type: string
  - Required: False

- **codingMode**
  - Type: string
  - Required: False

- **surroundMode**
  - Type: string
  - Required: False

- **attenuationControl**
  - Type: string
  - Required: False

- **lfeFilter**
  - Type: string
  - Required: False

- **phaseControl**
  - Type: string
  - Required: False

- **ltRtCenterMixLevel**
  - Left total/Right total center mix level. Only used for 3/2 coding mode. Valid values: 3.0, 1.5, 0.0, -1.5 -3.0 -4.5 -6.0 -60
  - Type: number
  - Required: False
**Properties**

**Format**
- Type: float
- Minimum: -60.0
- Maximum: 3.0

**dcFilter**
- Type: string
- Required: False

**bitstreamMode**
- Type: string
- Required: False

**stereoDownmix**
- Type: string
- Required: False

**loRoSurroundMixLevel**
Left only/Right only surround mix level. Only used for 3/2 coding mode. Valid values: -1.5, -3.0, -4.5, -6.0, -60
- Type: number
- Required: False
- Format: float
- Minimum: -60.0
- Maximum: -1.5

**loRoCenterMixLevel**
Left only/Right only center mix level. Only used for 3/2 coding mode. Valid values: 3.0, 1.5, 0.0, -1.5, -3.0, -4.5, -6.0, -60
- Type: number
- Required: False
- Format: float
- Minimum: -60.0
- Maximum: 3.0

**Eac3StereoDownmix (enum)**
Stereo downmix preference. Only used for 3/2 coding mode.

- NOT_INDICATED
- LO_RO
- LT_RT
- DPL2

**Eac3SurroundExMode (enum)**
When encoding 3/2 audio, sets whether an extra center back surround channel is matrix encoded into the left and right surround channels.
Eac3SurroundMode (enum)
When encoding 2/0 audio, sets whether Dolby Surround is matrix encoded into the two channels.

- NOT_INDICATED
- ENABLED
- DISABLED

EmbeddedConvert608To708 (enum)
When set to UPCONVERT, 608 data is both passed through via the "608 compatibility bytes" fields of the 708 wrapper as well as translated into 708. 708 data present in the source content will be discarded.

- UPCONVERT
- DISABLED

EmbeddedSourceSettings

source608ChannelNumber
Specifies the 608/708 channel number within the video track from which to extract captions. Unused for passthrough.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 4

convert608To708

- **Type:** string
- **Required:** False

source608TrackNumber
Specifies the video track index used for extracting captions. The system only supports one input video track, so this should always be set to '1'.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 1

ExceptionBody

message

- **Type:** string
**Properties**

**Required**: False

**F4vMoovPlacement (enum)**
If set to PROGRESSIVE_DOWNLOAD, the MOOV atom is relocated to the beginning of the archive as required for progressive downloading. Otherwise it is placed normally at the end.

- PROGRESSIVE_DOWNLOAD
- NORMAL

**F4vSettings**

**moovPlacement**

- **Type**: string
- **Required**: False

**FileGroupSettings**

**destination**

Use Destination (Destination) to specify the S3 output location and the output filename base. Destination accepts format identifiers. If you do not specify the base filename in the URI, the service will use the filename of the input file. If your job has multiple inputs, the service uses the filename of the first input file.

- **Type**: string
- **Required**: False
- **Pattern**: ^s3:\/\/

**FileSourceConvert608To708 (enum)**

If set to UPCONVERT, 608 caption data is both passed through via the "608 compatibility bytes" fields of the 708 wrapper as well as translated into 708. 708 data present in the source content will be discarded.

- UPCONVERT
- DISABLED

**FileSourceSettings**

**timeDelta**

Specifies a time delta in seconds to offset the captions from the source file.

- **Type**: integer
- **Required**: False
- **Minimum**: -2147483648
- **Maximum**: 2147483647

**convert608To708**

- **Type**: string
### SourceFile

External caption file used for loading captions. Accepted file extensions are 'scc', 'ttml', 'dfxp', 'stl', 'srt', and 'smi'.

- **Type:** string
- **Required:** True
- **Pattern:** `^(s3://)(.*?)\.(scc|SCC|ttml|TTML|dfxp|DFXP|stl|STL|srt|SRT|smi|SMI)\$`

### FrameCaptureSettings

#### framerateDenominator

Frame capture will encode the first frame of the output stream, then one frame every `framerateDenominator/framerateNumerator` seconds. For example, settings of `framerateNumerator = 1` and `framerateDenominator = 3` (a rate of 1/3 frame per second) will capture the first frame, then 1 frame every 3s. Files will be named as `filename.n.jpg` where `n` is the 0-based sequence number of each Capture.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 2147483647

#### maxCaptures

Maximum number of captures (encoded jpg output files).

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 10000000

#### framerateNumerator

Frame capture will encode the first frame of the output stream, then one frame every `framerateDenominator/framerateNumerator` seconds. For example, settings of `framerateNumerator = 1` and `framerateDenominator = 3` (a rate of 1/3 frame per second) will capture the first frame, then 1 frame every 3s. Files will be named as `filename.NNNNNNN.jpg` where `N` is the 0-based frame sequence number zero padded to 7 decimal places.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 2147483647

#### quality

JPEG Quality - a higher value equals higher quality.

- **Type:** integer
Required: False
Minimum: 1
Maximum: 100

**H264AdaptiveQuantization (enum)**
Adaptive quantization. Allows intra-frame quantizers to vary to improve visual quality.
- OFF
- LOW
- MEDIUM
- HIGH
- HIGHER
- MAX

**H264CodecLevel (enum)**
H.264 Level.
- AUTO
- LEVEL_1
- LEVEL_1_1
- LEVEL_1_2
- LEVEL_1_3
- LEVEL_2
- LEVEL_2_1
- LEVEL_2_2
- LEVEL_3
- LEVEL_3_1
- LEVEL_3_2
- LEVEL_4
- LEVEL_4_1
- LEVEL_4_2
- LEVEL_5
- LEVEL_5_1
- LEVEL_5_2

**H264CodecProfile (enum)**
H.264 Profile. High 4:2:2 and 10-bit profiles are only available with the AVC-I License.
- BASELINE
- HIGH
- HIGH_10BIT
- HIGH_422
- HIGH_422_10BIT
- MAIN

**H264EntropyEncoding (enum)**
Entropy encoding mode. Use CABAC (must be in Main or High profile) or CAVLC.
CABAC
CAVLC

H264FieldEncoding (enum)

Choosing FORCE_FIELD disables PAFF encoding for interlaced outputs.

PAFF
FORCE_FIELD

H264FlickerAdaptiveQuantization (enum)

Adjust quantization within each frame to reduce flicker or 'pop' on I-frames.

DISABLED
ENABLED

H264FramerateControl (enum)

If you are using the console, use the Framerate setting to specify the framerate for this output. If you want to keep the same framerate as the input video, choose Follow source. If you want to do framerate conversion, choose a framerate from the dropdown list or choose Custom. The framerates shown in the dropdown list are decimal approximations of fractions. If you choose Custom, specify your framerate as a fraction. If you are creating your transcoding job specification as a JSON file without the console, use FramerateControl to specify which value the service uses for the framerate for this output. Choose INITIALIZE_FROM_SOURCE if you want the service to use the framerate from the input. Choose SPECIFIED if you want the service to use the framerate you specify in the settings FramerateNumerator and FramerateDenominator.

INITIALIZE_FROM_SOURCE
SPECIFIED

H264FramerateConversionAlgorithm (enum)

When set to INTERPOLATE, produces smoother motion during framerate conversion.

DUPLICATE_DROP
INTERPOLATE

H264GopBReference (enum)

If enable, use reference B frames for GOP structures that have B frames > 1.

DISABLED
ENABLED

H264GopSizeUnits (enum)

Indicates if the GOP Size in H264 is specified in frames or seconds. If seconds the system will convert the GOP Size into a frame count at run time.

FRAMES
SECONDS
H264InterlaceMode (enum)

Use Interlace mode (InterlaceMode) to choose the scan line type for the output. * Top Field First (TOP_FIELD) and Bottom Field First (BOTTOM_FIELD) produce interlaced output with the entire output having the same field polarity (top or bottom first). * Follow, Default Top (FOLLOW_TOP_FIELD) and Follow, Default Bottom (FOLLOW_BOTTOM_FIELD) use the same field polarity as the source. Therefore, behavior depends on the input scan type, as follows. - If the source is interlaced, the output will be interlaced with the same polarity as the source (it will follow the source). The output could therefore be a mix of “top field first” and “bottom field first”. - If the source is progressive, the output will be interlaced with “top field first” or “bottom field first” polarity, depending on which of the Follow options you chose.

PROGRESSIVE
TOP_FIELD
BOTTOM_FIELD
FOLLOW_TOP_FIELD
FOLLOW_BOTTOM_FIELD

H264ParControl (enum)

Using the API, enable ParFollowSource if you want the service to use the pixel aspect ratio from the input. Using the console, do this by choosing Follow source for Pixel aspect ratio.

INITIALIZE_FROM_SOURCE
SPECIFIED

H264QualityTuningLevel (enum)

Use Quality tuning level (H264QualityTuningLevel) to specify whether to use fast single-pass, high-quality singlepass, or high-quality multipass video encoding.

SINGLE_PASS
SINGLE_PASS_HQ
MULTI_PASS_HQ

H264RateControlMode (enum)

Use this setting to specify whether this output has a variable bitrate (VBR) or constant bitrate (CBR).

VBR
CBR

H264RepeatPps (enum)

Places a PPS header on each encoded picture, even if repeated.

DISABLED
ENABLED

H264SceneChangeDetect (enum)

Scene change detection (inserts I-frames on scene changes).

DISABLED
ENABLED
H264Settings

slices

Number of slices per picture. Must be less than or equal to the number of macroblock rows for progressive pictures, and less than or equal to half the number of macroblock rows for interlaced pictures.

Type: integer
Required: False
Minimum: 1
Maximum: 32

minIInterval

Enforces separation between repeated (cadence) I-frames and I-frames inserted by Scene Change Detection. If a scene change I-frame is within I-interval frames of a cadence I-frame, the GOP is shrunk and/or stretched to the scene change I-frame. GOP stretch requires enabling lookahead as well as setting I-interval. The normal cadence resumes for the next GOP. This setting is only used when Scene Change Detect is enabled. Note: Maximum GOP stretch = GOP size + Min-I-interval - 1

Type: integer
Required: False
Minimum: 0
Maximum: 30

parNumerator

Pixel Aspect Ratio numerator.

Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

flickerAdaptiveQuantization

Type: string
Required: False

gopSizeUnits

Type: string
Required: False

hrdBufferSize

Size of buffer (HRD buffer model) in bits. For example, enter five megabits as 5000000.

Type: integer
Required: False
Minimum: 0
Maximum: 1152000000

qualityTuningLevel

Type: string
Required: False

maxBitrate

Maximum bitrate in bits/second. For example, enter five megabits per second as 5000000.

Type: integer
Required: False
Minimum: 1000
Maximum: 1152000000

bitrate

Average bitrate in bits/second. Required for VBR and CBR. For MS Smooth outputs, bitrates must be unique when rounded down to the nearest multiple of 1000.

Type: integer
Required: False
Minimum: 1000
Maximum: 1152000000

spatialAdaptiveQuantization

Type: string
Required: False

slowPal

Type: string
Required: False

codecProfile

Type: string
Required: False

unregisteredSeiTimecode

Type: string
Required: False

softness

Softness. Selects quantizer matrix, larger values reduce high-frequency content in the encoded image.

Type: integer
Properties

framerateControl
Type: string
Required: False

telecine
Type: string
Required: False

codecLevel
Type: string
Required: False

framerateConversionAlgorithm
Type: string
Required: False

numberReferenceFrames
Number of reference frames to use. The encoder may use more than requested if using B-frames and/or interlaced encoding.
Type: integer
Required: False
Minimum: 1
Maximum: 6

temporalAdaptiveQuantization
Type: string
Required: False

repeatPps
Type: string
Required: False

hrdBufferInitialFillPercentage
Percentage of the buffer that should initially be filled (HRD buffer model).
Type: integer
Required: False
Minimum: 0
Maximum: 100

framerateNumerator

Framerate numerator - framerate is a fraction, e.g. 24000 / 1001 = 23.976 fps.

   Type: integer
   Required: False
   Minimum: 1
   Maximum: 2147483647

gopClosedCadence

Frequency of closed GOPs. In streaming applications, it is recommended that this be set to 1 so a decoder joining mid-stream will receive an IDR frame as quickly as possible. Setting this value to 0 will break output segmenting.

   Type: integer
   Required: False
   Minimum: 0
   Maximum: 2147483647

numberBFramesBetweenReferenceFrames

Number of B-frames between reference frames.

   Type: integer
   Required: False
   Minimum: 0
   Maximum: 7

framerateDenominator

When you use the API for transcoding jobs that use framerate conversion, specify the framerate as a fraction. For example, 24000 / 1001 = 23.976 fps. Use FramerateDenominator to specify the denominator of this fraction. In this example, use 1001 for the value of FramerateDenominator. When you use the console for transcoding jobs that use framerate conversion, provide the value as a decimal number for Framerate. In this example, specify 23.976.

   Type: integer
   Required: False
   Minimum: 1
   Maximum: 2147483647

entropyEncoding

   Type: string
   Required: False

fieldEncoding

   Type: string
   Required: False
adaptiveQuantization
Type: string
Required: False

interlaceMode
Type: string
Required: False

gopSize
GOP Length (keyframe interval) in frames or seconds. Must be greater than zero.
Type: number
Required: False
Format: float
Minimum: 0.0

parDenominator
Pixel Aspect Ratio denominator.
Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

sceneChangeDetect
Type: string
Required: False

parControl
Type: string
Required: False

rateControlMode
Type: string
Required: False
**H264SlowPal (enum)**

Enables Slow PAL rate conversion. 23.976fps and 24fps input is relabeled as 25fps, and audio is sped up correspondingly.

- DISABLED
- ENABLED

**H264SpatialAdaptiveQuantization (enum)**

Adjust quantization within each frame based on spatial variation of content complexity.

- DISABLED
- ENABLED

**H264Syntax (enum)**

Produces a bitstream compliant with SMPTE RP-2027.

- DEFAULT
- RP2027

**H264Telecine (enum)**

This field applies only if the Streams > Advanced > Framerate (framerate) field is set to 29.970. This field works with the Streams > Advanced > Preprocessors > Deinterlacer field (deinterlace_mode) and the Streams > Advanced > Interlaced Mode field (interlace_mode) to identify the scan type for the output: Progressive, Interlaced, Hard Telecine or Soft Telecine. - Hard: produces 29.97i output from 23.976 input. - Soft: produces 23.976; the player converts this output to 29.97i.

- NONE
- SOFT
- HARD

**H264TemporalAdaptiveQuantization (enum)**

Adjust quantization within each frame based on temporal variation of content complexity.

- DISABLED
- ENABLED

**H264UnregisteredSeiTimecode (enum)**

Inserts timecode for each frame as 4 bytes of an unregistered SEI message.

- DISABLED
- ENABLED

**H265AdaptiveQuantization (enum)**

Adaptive quantization. Allows intra-frame quantizers to vary to improve visual quality.

- OFF
- LOW
**Properties**

**H265AlternateTransferFunctionSei (enum)**

Enables Alternate Transfer Function SEI message for outputs using Hybrid Log Gamma (HLG) Electro-Optical Transfer Function (EOTF).

- DISABLED
- ENABLED

**H265CodecLevel (enum)**

H.265 Level.

- AUTO
- LEVEL_1
- LEVEL_2
- LEVEL_2_1
- LEVEL_3
- LEVEL_3_1
- LEVEL_4
- LEVEL_4_1
- LEVEL_5
- LEVEL_5_1
- LEVEL_5_2
- LEVEL_6
- LEVEL_6_1
- LEVEL_6_2

**H265CodecProfile (enum)**

Represents the Profile and Tier, per the HEVC (H.265) specification. Selections are grouped as [Profile] / [Tier], so “Main/High” represents Main Profile with High Tier. 4:2:2 profiles are only available with the HEVC 4:2:2 License.

- MAIN_MAIN
- MAIN_HIGH
- MAIN10_MAIN
- MAIN10_HIGH
- MAIN_422_8BIT_MAIN
- MAIN_422_8BIT_HIGH
- MAIN_422_10BIT_MAIN
- MAIN_422_10BIT_HIGH

**H265FlickerAdaptiveQuantization (enum)**

Adjust quantization within each frame to reduce flicker or ‘pop’ on I-frames.

- DISABLED
ENABLED

**H265FramerateControl (enum)**

If you are using the console, use the Framerate setting to specify the framerate for this output. If you want to keep the same framerate as the input video, choose Follow source. If you want to do framerate conversion, choose a framerate from the dropdown list or choose Custom. The framerates shown in the dropdown list are decimal approximations of fractions. If you choose Custom, specify your framerate as a fraction. If you are creating your transcoding job specification as a JSON file without the console, use FramerateControl to specify which value the service uses for the framerate for this output. Choose INITIALIZE_FROM_SOURCE if you want the service to use the framerate from the input. Choose SPECIFIED if you want the service to use the framerate you specify in the settings FramerateNumerator and FramerateDenominator.

  INITIALIZE_FROM_SOURCE
  SPECIFIED

**H265FramerateConversionAlgorithm (enum)**

When set to INTERPOLATE, produces smoother motion during framerate conversion.

  DUPPLICATE_DROP
  INTERPOLATE

**H265GopBReference (enum)**

If enable, use reference B frames for GOP structures that have B frames > 1.

  DISABLED
  ENABLED

**H265GopSizeUnits (enum)**

Indicates if the GOP Size in H265 is specified in frames or seconds. If seconds the system will convert the GOP Size into a frame count at run time.

  FRAMES
  SECONDS

**H265InterlaceMode (enum)**

Use Interlace mode (InterlaceMode) to choose the scan line type for the output. * Top Field First (TOP_FIELD) and Bottom Field First (BOTTOM_FIELD) produce interlaced output with the entire output having the same field polarity (top or bottom first). * Follow, Default Top (FOLLOW_TOP_FIELD) and Follow, Default Bottom (FOLLOW_BOTTOM_FIELD) use the same field polarity as the source. Therefore, behavior depends on the input scan type. - If the source is interlaced, the output will be interlaced with the same polarity as the source (it will follow the source). The output could therefore be a mix of “top field first” and “bottom field first”. - If the source is progressive, the output will be interlaced with “top field first” or “bottom field first” polarity, depending on which of the Follow options you chose.

  PROGRESSIVE
  TOP_FIELD
  BOTTOM_FIELD
  FOLLOW_TOP_FIELD
H265ParControl (enum)
Using the API, enable ParFollowSource if you want the service to use the pixel aspect ratio from the input. Using the console, do this by choosing Follow source for Pixel aspect ratio.

INITIALIZE_FROM_SOURCE
SPECIFIED

H265QualityTuningLevel (enum)
Use Quality tuning level (H265QualityTuningLevel) to specify whether to use fast single-pass, high-quality singlepass, or high-quality multipass video encoding.

SINGLE_PASS
SINGLE_PASS_HQ
MULTI_PASS_HQ

H265RateControlMode (enum)
Use this setting to specify whether this output has a variable bitrate (VBR) or constant bitrate (CBR).

VBR
CBR

H265SampleAdaptiveOffsetFilterMode (enum)
Specify Sample Adaptive Offset (SAO) filter strength. Adaptive mode dynamically selects best strength based on content.

DEFAULT
ADAPTIVE
OFF

H265SceneChangeDetect (enum)
Scene change detection (inserts I-frames on scene changes).

DISABLED
ENABLED

H265Settings
slices
Number of slices per picture. Must be less than or equal to the number of macroblock rows for progressive pictures, and less than or equal to half the number of macroblock rows for interlaced pictures.

Type: integer
Required: False
Minimum: 1
Minimum Interval

Enforces separation between repeated (cadence) I-frames and I-frames inserted by Scene Change Detection. If a scene change I-frame is within I-interval frames of a cadence I-frame, the GOP is shrunk and/or stretched to the scene change I-frame. GOP stretch requires enabling lookahead as well as setting I-interval. The normal cadence resumes for the next GOP. This setting is only used when Scene Change Detect is enabled. Note: Maximum GOP stretch = GOP size + Min-I-interval - 1

Type: integer
Required: False
Minimum: 0
Maximum: 30

parNumerator

Pixel Aspect Ratio numerator.

Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

flickerAdaptiveQuantization

Type: string
Required: False

gopSizeUnits

Type: string
Required: False

hrdBufferSize

Size of buffer (HRD buffer model) in bits. For example, enter five megabits as 5000000.

Type: integer
Required: False
Minimum: 0
Maximum: 1466400000

qualityTuningLevel

Type: string
Required: False

maxBitrate

Maximum bitrate in bits/second.

Type: integer
Required: False
Properties

Minimum: 1000
Maximum: 1466400000

bitrate
Average bitrate in bits/second. Required for VBR and CBR. For MS Smooth outputs, bitrates must be unique when rounded down to the nearest multiple of 1000.

Type: integer
Required: False
Minimum: 1000
Maximum: 1466400000

spatialAdaptiveQuantization
Type: string
Required: False

temporalIds
Type: string
Required: False

slowPal
Type: string
Required: False

tiles
Type: string
Required: False

writeMp4PackagingType
Type: string
Required: False

codecProfile
Type: string
Required: False

alternateTransferFunctionSei
Type: string
Required: False
unregisteredSeiTImecode
  Type: string
  Required: False

framerateControl
  Type: string
  Required: False

telecine
  Type: string
  Required: False

codecLevel
  Type: string
  Required: False

framerateConversionAlgorithm
  Type: string
  Required: False

numberReferenceFrames
Number of reference frames to use. The encoder may use more than requested if using B-frames and/or interlaced encoding.
  Type: integer
  Required: False
  Minimum: 1
  Maximum: 6

temporalAdaptiveQuantization
  Type: string
  Required: False

hrdBufferInitialFillPercentage
Percentage of the buffer that should initially be filled (HRD buffer model).
  Type: integer
  Required: False
  Minimum: 0
  Maximum: 100

framerateNumerator
Framerate numerator - framerate is a fraction, e.g. 24000 / 1001 = 23.976 fps.
Properties

**gopClosedCadence**

Frequency of closed GOPs. In streaming applications, it is recommended that this be set to 1 so a decoder joining mid-stream will receive an IDR frame as quickly as possible. Setting this value to 0 will break output segmenting.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 2147483647

**numberBFramesBetweenReferenceFrames**

Number of B-frames between reference frames.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
- **Maximum:** 7

**framerateDenominator**

Framerate denominator.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 2147483647

**adaptiveQuantization**

- **Type:** string
- **Required:** False

**interlaceMode**

- **Type:** string
- **Required:** False

**gopSize**

GOP Length (keyframe interval) in frames or seconds. Must be greater than zero.

- **Type:** number
- **Required:** False
- **Format:** float
- **Minimum:** 0.0

---

440
Properties

**gopBReference**
- Type: string
- Required: False

**parDenominator**
Pixel Aspect Ratio denominator.
- Type: integer
- Required: False
- Minimum: 1
- Maximum: 2147483647

**sceneChangeDetect**
- Type: string
- Required: False

**parControl**
- Type: string
- Required: False

**rateControlMode**
- Type: string
- Required: False

**H265SlowPal (enum)**
Enables Slow PAL rate conversion. 23.976fps and 24fps input is relabeled as 25fps, and audio is sped up correspondingly.

- DISABLED
- ENABLED

**H265SpatialAdaptiveQuantization (enum)**
Adjust quantization within each frame based on spatial variation of content complexity.

- DISABLED
- ENABLED

**H265Telecine (enum)**
This field applies only if the Streams > Advanced > Framerate (framerate) field is set to 29.970. This field works with the Streams > Advanced > Preprocessors > Deinterlacer field (deinterlace_mode) and the Streams > Advanced > Interlaced Mode field (interlace_mode) to identify the scan type for the output:
- Progressive, Interlaced, Hard Telecine or Soft Telecine.
- Hard: produces 29.97i output from 23.976 input.
- Soft: produces 23.976; the player converts this output to 29.97i.

- NONE
- SOFT
HARD

**H265TemporalAdaptiveQuantization (enum)**

Adjust quantization within each frame based on temporal variation of content complexity.

- DISABLED
- ENABLED

**H265TemporalIds (enum)**

Enables temporal layer identifiers in the encoded bitstream. Up to 3 layers are supported depending on GOP structure: I- and P-frames form one layer, reference B-frames can form a second layer and non-reference b-frames can form a third layer. Decoders can optionally decode only the lower temporal layers to generate a lower frame rate output. For example, given a bitstream with temporal IDs and with b-frames = 1 (i.e. IbPbPb display order), a decoder could decode all the frames for full frame rate output or only the I and P frames (lowest temporal layer) for a half frame rate output.

- DISABLED
- ENABLED

**H265Tiles (enum)**

Enable use of tiles, allowing horizontal as well as vertical subdivision of the encoded pictures.

- DISABLED
- ENABLED

**H265UnregisteredSeiTimecode (enum)**

Inserts timecode for each frame as 4 bytes of an unregistered SEI message.

- DISABLED
- ENABLED

**H265WriteMp4PackagingType (enum)**

If HVC1, output that is H.265 will be marked as HVC1 and adhere to the ISO-IECJTC1-SC29_N13798_Text_ISOIEC_FDIS_14496-15_3rd_E spec which states that parameter set NAL units will be stored in the sample headers but not in the samples directly. If HEV1, then H.265 will be marked as HEV1 and parameter set NAL units will be written into the samples.

- HVC1
- HEV1

**Hdr10Metadata**

**redPrimaryY**

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

**Type:** integer
**greenPrimaryY**

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 50000

**whitePointX**

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 50000

**maxLuminance**

Nominal maximum mastering display luminance in units of 0.0001 candelas per square meter.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 2147483647

**greenPrimaryX**

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 50000

**whitePointY**

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

- **Type**: integer
- **Required**: False
**redPrimaryX**

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 50000

**bluePrimaryX**

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 50000

**bluePrimaryY**

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 50000

**maxFrameAverageLightLevel**

Maximum average light level of any frame in the coded video sequence, in units of candelas per square meter.

- **Type**: integer
- **Required**: True
- **Minimum**: 0
- **Maximum**: 65535

**maxContentLightLevel**

Maximum light level among all samples in the coded video sequence, in units of candelas per square meter.

- **Type**: integer
- **Required**: True
- **Minimum**: 0
**Properties**

**Minimum**: 65535

**minLuminance**
Nominal minimum mastering display luminance in units of 0.0001 candelas per square meter

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 2147483647

**HlsMarkers (enum)**

- ELEMENTAL
- ELEMENTAL_SCTE35

**HlsAudioTrackType (enum)**

Four types of audio-only tracks are supported:

- **Audio-Only Variant Stream**
  The client can play back this audio-only stream instead of video in low-bandwidth scenarios. Represented as an EXT-X-STREAM-INF in the HLS manifest. Alternate Audio, Auto Select, Default Alternate rendition that the client should try to play back by default. Represented as an EXT-X-MEDIA in the HLS manifest with DEFAULT=YES, AUTOSELECT=YES
- **Alternate Audio, Auto Select, Default**
  Alternate rendition that the client should try to play back by default. Represented as an EXT-X-MEDIA in the HLS manifest with DEFAULT=YES, AUTOSELECT=YES
- **Alternate Audio, Auto Select, Not Default**
  Alternate rendition that the client may try to play back by default. Represented as an EXT-X-MEDIA in the HLS manifest with DEFAULT=NO, AUTOSELECT=YES
- **Alternate Audio, not Auto Select**
  Alternate rendition that the client will not try to play back by default. Represented as an EXT-X-MEDIA in the HLS manifest with DEFAULT=NO, AUTOSELECT=NO

- ALTERNATE_AUDIO_AUTO_SELECT_DEFAULT
- ALTERNATE_AUDIO_AUTO_SELECT
- ALTERNATE_AUDIO_NOT_AUTO_SELECT
- AUDIO_ONLY_VARIANT_STREAM

**HlsCaptionLanguageMapping**

**languageDescription**
Caption language description.

- **Type**: string
- **Required**: False

**captionChannel**
Caption channel.

- **Type**: integer
- **Required**: False
- **Minimum**: -2147483648
- **Maximum**: 2147483647

**languageCode**

- **Type**: string
**Required:** False

**customLanguageCode**

Specify the language for this caption channel, using the ISO 639-2 or ISO 639-3 three-letter language code.

**Type:** string
**Required:** False
**Pattern:** ^[A-Za-z]{3}$

**HlsCaptionLanguageSetting (enum)**

Applies only to 608 Embedded output captions. Insert: Include CLOSED-CAPTIONS lines in the manifest. Specify at least one language in the CC1 Language Code field. One CLOSED-CAPTION line is added for each Language Code you specify. Make sure to specify the languages in the order in which they appear in the original source (if the source is embedded format) or the order of the caption selectors (if the source is other than embedded). Otherwise, languages in the manifest will not match up properly with the output captions. None: Include CLOSED-CAPTIONS=NONE line in the manifest. Omit: Omit any CLOSED-CAPTIONS line from the manifest.

- INSERT
- OMIT
- NONE

**HlsClientCache (enum)**

When set to ENABLED, sets #EXT-X-ALLOW-CACHE:no tag, which prevents client from saving media segments for later replay.

- DISABLED
- ENABLED

**HlsCodecSpecification (enum)**

Specification to use (RFC-6381 or the default RFC-4281) during m3u8 playlist generation.

- RFC_6381
- RFC_4281

**HlsDirectoryStructure (enum)**

Indicates whether segments should be placed in subdirectories.

- SINGLE_DIRECTORY
- SUBDIRECTORY_PER_STREAM

**HlsEncryptionSettings**

**initializationVectorInManifest**

**Type:** string
**Required:** False
constantInitializationVector

This is a 128-bit, 16-byte hex value represented by a 32-character text string. If this parameter is not set then the Initialization Vector will follow the segment number by default.

Type: string
Required: False
Pattern: ^[0-9a-fA-F]{32}$

staticKeyProvider

Type: StaticKeyProvider (p. 497)
Required: False

type

Type: string
Required: True

encryptionMethod

Type: string
Required: False

spekeKeyProvider

Type: SpekeKeyProvider (p. 496)
Required: False

HlsEncryptionType (enum)

Encrypts the segments with the given encryption scheme. Leave blank to disable. Selecting 'Disabled' in the web interface also disables encryption.

AES128
SAMPLE_AES

HlsGroupSettings

segmentsPerSubdirectory

Number of segments to write to a subdirectory before starting a new one. directoryStructure must be SINGLE_DIRECTORY for this setting to have an effect.

Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

streamInfResolution

Type: string
Required: False

timestampDeltaMilliseconds

Provides an extra millisecond delta offset to fine tune the timestamps.

Type: integer
Required: False
Minimum: -2147483648
Maximum: 2147483647

outputSelection

Type: string
Required: False

captionLanguageMappings

Language to be used on Caption outputs

Type: Array of type HlsCaptionLanguageMapping (p. 445)
Required: False

clientCache

Type: string
Required: False

codecSpecification

Type: string
Required: False

destination

Use Destination (Destination) to specify the S3 output location and the output filename base. Destination accepts format identifiers. If you do not specify the base filename in the URI, the service will use the filename of the input file. If your job has multiple inputs, the service uses the filename of the first input file.

Type: string
Required: False
Pattern: ^s3:/\/

timedMetadataId3Frame

Type: string
Required: False

segmentControl

Type: string
Required: False
timedMetadataId3Period

Timed Metadata interval in seconds.

Type: integer
Required: False
Minimum: -2147483648
Maximum: 2147483647

captionLanguageSetting

Type: string
Required: False

minSegmentLength

When set, Minimum Segment Size is enforced by looking ahead and back within the specified range for a nearby avail and extending the segment size if needed.

Type: integer
Required: True
Minimum: 0
Maximum: 2147483647

directoryStructure

Type: string
Required: False

programDateTime

Type: string
Required: False

baseUrl

A partial URI prefix that will be prepended to each output in the media .m3u8 file. Can be used if base manifest is delivered from a different URL than the main .m3u8 file.

Type: string
Required: False

encryption

DRM settings.

Type: HlsEncryptionSettings (p. 446)
Required: False

adMarkers

Choose one or more ad marker types to pass SCTE35 signals through to this group of Apple HLS outputs.
Type: Array of type string
Required: False

programDateTimePeriod

Period of insertion of EXT-X-PROGRAM-DATE-TIME entry, in seconds.

Type: integer
Required: False
Minimum: 0
Maximum: 3600

manifestCompression

Type: string
Required: False

segmentLength

Length of MPEG-2 Transport Stream segments to create (in seconds). Note that segments will end on the next keyframe after this number of seconds, so actual segment length may be longer.

Type: integer
Required: True
Minimum: 1
Maximum: 2147483647

manifestDurationFormat

Type: string
Required: False

HlsIFrameOnlyManifest (enum)

When set to INCLUDE, writes I-Frame Only Manifest in addition to the HLS manifest

INCLUDE
EXCLUDE

HlsInitializationVectorInManifest (enum)

The Initialization Vector is a 128-bit number used in conjunction with the key for encrypting blocks. If set to INCLUDE, Initialization Vector is listed in the manifest. Otherwise Initialization Vector is not in the manifest.

INCLUDE
EXCLUDE

HlsKeyProviderType (enum)

Indicates which type of key provider is used for encryption.

SPEKE
Properties

**STATIC_KEY**

**HlsManifestCompression (enum)**
When set to GZIP, compresses HLS playlist.

- GZIP
- NONE

**HlsManifestDurationFormat (enum)**
Indicates whether the output manifest should use floating point values for segment duration.

- FLOATING_POINT
- INTEGER

**HlsOutputSelection (enum)**
Indicates whether the .m3u8 manifest file should be generated for this HLS output group.

- MANIFESTS_AND_SEGMENTS
- SEGMENTS_ONLY

**HlsProgramDateTime (enum)**
Includes or excludes EXT-X-PROGRAM-DATE-TIME tag in .m3u8 manifest files. The value is calculated as follows: either the program date and time are initialized using the input timecode source, or the time is initialized using the input timecode source and the date is initialized using the timestamp_offset.

- INCLUDE
- EXCLUDE

**HlsSegmentControl (enum)**
When set to SINGLE_FILE, emits program as a single media resource (.ts) file, uses #EXT-X-BYTERANGE tags to index segment for playback.

- SINGLE_FILE
- SEGMENTED_FILES

**HlsSettings**

**iFrameOnlyManifest**

- **Type:** string
- **Required:** False

**audioRenditionSets**

List all the audio groups that are used with the video output stream. Input all the audio GROUP-IDs that are associated to the video, separate by ",".

- **Type:** string
- **Required:** False
### Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>audioTrackType</td>
<td>string</td>
<td>False</td>
<td>Type: string Required: False</td>
</tr>
<tr>
<td>segmentModifier</td>
<td>string</td>
<td>False</td>
<td>Type: string Required: False</td>
</tr>
<tr>
<td>audioGroupId</td>
<td>string</td>
<td>False</td>
<td>Type: string Required: False</td>
</tr>
<tr>
<td>HlsStreamInfResolution</td>
<td>enum</td>
<td></td>
<td>Include or exclude RESOLUTION attribute for video in EXT-X-STREAM-INF tag</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>of variant manifest.</td>
</tr>
<tr>
<td></td>
<td>INCLUDE</td>
<td></td>
<td>INCLUDE</td>
</tr>
<tr>
<td></td>
<td>EXCLUDE</td>
<td></td>
<td>EXCLUDE</td>
</tr>
<tr>
<td>HlsTimedMetadataId3Frame</td>
<td>enum</td>
<td></td>
<td>Indicates ID3 frame that has the timecode.</td>
</tr>
<tr>
<td></td>
<td>NONE</td>
<td></td>
<td>NONE</td>
</tr>
<tr>
<td></td>
<td>PRIV</td>
<td></td>
<td>PRIV</td>
</tr>
<tr>
<td></td>
<td>TDRL</td>
<td></td>
<td>TDRL</td>
</tr>
<tr>
<td>Id3Insertion</td>
<td></td>
<td></td>
<td>Use ID3 tag (Id3) to provide a tag value in base64-encode format.</td>
</tr>
<tr>
<td>id3</td>
<td>string</td>
<td>True</td>
<td>Type: string Required: True Pattern: ^[A-Za-z0-9+/]+={0,2}$</td>
</tr>
<tr>
<td></td>
<td>string</td>
<td>True</td>
<td>Type: string Required: True Format: timecode Pattern: ^([01][0-9]</td>
</tr>
</tbody>
</table>
**ImageInserter**

**insertableImages**

Image to insert. Must be 32 bit windows BMP, PNG, or TGA file. Must not be larger than the output frames.

*Type:* Array of type `InsertableImage (p. 456)`
*Required:* True

**Input**

**audioSelectors**

Use Audio selectors (AudioSelectors) to specify a track or set of tracks from the input that you will use in your outputs. You can use multiple Audio selectors per input.

*Type:* object
*Required:* False

**audioSelectorGroups**

Specifies set of audio selectors within an input to combine. An input may have multiple audio selector groups. See "Audio Selector Group":#inputs-audio_selector_group for more information.

*Type:* object
*Required:* False

**videoSelector**

*Type:* `VideoSelector (p. 505)`
*Required:* False

**denoiseFilter**

*Type:* string
*Required:* False

**filterEnable**

*Type:* string
*Required:* False

**deblockFilter**

*Type:* string
*Required:* False

**filterStrength**

Use Filter strength (FilterStrength) to adjust the magnitude the input filter settings (Deblock and Denoise). The range is -5 to 5. Default is 0.
**programNumber**

Use Program (programNumber) to select a specific program from within a multi-program transport stream. Note that Quad 4K is not currently supported. Default is the first program within the transport stream. If the program you specify doesn’t exist, the transcoding service will use this default.

- **Type**: integer
- **Required**: False
- **Minimum**: -5
- **Maximum**: 5

**timecodeSource**

- **Type**: string
- **Required**: False

**captionSelectors**

Use Captions selectors (CaptionSelectors) to specify the captions data from the input that you will use in your outputs. You can use multiple captions selectors per input.

- **Type**: object
- **Required**: False

**fileInput**

Use Input (fileInput) to define the source file used in the transcode job. There can be multiple inputs in a job. These inputs are concatenated, in the order they are specified in the job, to create the output.

- **Type**: string
- **Required**: True
- **Pattern**: ^s3:\/[^/\]+/\+([^/\])+([^/\]+)(((\w+)\d|\d)\d))$
Properties

Required: False

psiControl

Type: string
Required: False

InputClipping

startTimecode

Set Start timecode (StartTimecode) to the beginning of the portion of the input you are clipping. The frame corresponding to the Start timecode value is included in the clip. Start timecode or End timecode may be left blank, but not both. Use the format HH:MM:SS:FF or HH:MM:SS;FF, where HH is the hour, MM is the minute, SS is the second, and FF is the frame number. When choosing this value, take into account your setting for Input timecode source. For example, if you have embedded timecodes that start at 01:00:00:00 and you want your clip to begin five minutes into the video, use 01:05:00:00.

Type: string
Required: False
Format: timecode
Pattern: ^([01][0-9]|2[0-4]):[0-5][0-9]:[0-5][0-9]:[0-9]{2}$

dendTimecode

Set End timecode (EndTimecode) to the end of the portion of the input you are clipping. The frame corresponding to the End timecode value is included in the clip. Start timecode or End timecode may be left blank, but not both. Use the format HH:MM:SS:FF or HH:MM:SS;FF, where HH is the hour, MM is the minute, SS is the second, and FF is the frame number. When choosing this value, take into account your setting for timecode source under input settings (InputTimecodeSource). For example, if you have embedded timecodes that start at 01:00:00:00 and you want your clip to end six minutes into the video, use 01:06:00:00.

Type: string
Required: False
Format: timecode
Pattern: ^([01][0-9]|2[0-4]):[0-5][0-9]:[0-5][0-9]:[0-9]{2}$

InputDeblockFilter (enum)

Enable Deblock (InputDeblockFilter) to produce smoother motion in the output. Default is disabled. Only manually controllable for MPEG2 and uncompressed video inputs.

ENABLED
DISABLED

InputDenoiseFilter (enum)

Enable Denoise (InputDenoiseFilter) to filter noise from the input. Default is disabled. Only applicable to MPEG2, H.264, H.265, and uncompressed video inputs.

ENABLED
DISABLED
InputFilterEnable (enum)

Use Filter enable (InputFilterEnable) to specify how the transcoding service applies the denoise and deblock filters. You must also enable the filters separately, with Denoise (InputDenoiseFilter) and Deblock (InputDeblockFilter). * Auto - The transcoding service determines whether to apply filtering, depending on input type and quality. * Disable - The input is not filtered. This is true even if you use the API to enable them in (InputDeblockFilter) and (InputDeblockFilter). * Force - The input is filtered regardless of input type.

AUTO
DISABLE
FORCE

InputPsiControl (enum)

Set PSI control (InputPsiControl) for transport stream inputs to specify which data the demux process to scan. * Ignore PSI - Scan all PIDs for audio and video. * Use PSI - Scan only PSI data.

IGNORE_PSI
USE_PSI

InputTimecodeSource (enum)

Timecode source under input settings (InputTimecodeSource) only affects the behavior of features that apply to a single input at a time, such as input clipping and synchronizing some captions formats. Use this setting to specify whether the service counts frames by timecodes embedded in the video (EMBEDDED) or by starting the first frame at zero (ZEROBASED). In both cases, the timecode format is HH:MM:SS:FF or HH:MM:SS;FF, where FF is the frame number. Only set this to EMBEDDED if your source video has embedded timecodes.

EMBEDDED
ZEROBASED
SPECIFIEDSTART

InsertableImage

duration

Use Duration (Duration) to set the time, in milliseconds, for the image to remain on the output video.

Type: integer
Required: False
Minimum: -2147483648
Maximum: 2147483647

fadeOut

Use Fade out (FadeOut) to set the length, in milliseconds, of the inserted image fade out. If you don’t specify a value for Fade out, the image will disappear abruptly at the end of the inserted image duration.

Type: integer
Required: False
Minimum: -2147483648
Maximum: 2147483647
Properties

imageY
Use Top (ImageY) to set the distance, in pixels, between the inserted image and the top edge of the video frame. Required for BMP, PNG and TGA input.

- **Type**: integer
- **Required**: True
- **Minimum**: -2147483648
- **Maximum**: 2147483647

fadeIn
Use Fade in (FadeIn) to set the length, in milliseconds, of the inserted image fade in. If you don't specify a value for Fade in, the image will appear abruptly at the Start time.

- **Type**: integer
- **Required**: False
- **Minimum**: -2147483648
- **Maximum**: 2147483647

imageX
Use Left (ImageX) to set the distance, in pixels, between the inserted image and the left edge of the frame. Required for BMP, PNG and TGA input.

- **Type**: integer
- **Required**: True
- **Minimum**: -2147483648
- **Maximum**: 2147483647

width
Specify the Width (Width) of the inserted image. Use a value that is less than or equal to the video resolution width. Leave this setting blank to use the native width of the image.

- **Type**: integer
- **Required**: False
- **Minimum**: -2147483648
- **Maximum**: 2147483647

startTime
Use Start time (StartTime) to specify the video timecode when the image is inserted in the output. This must be in timecode (HH:MM:SS:FF or HH:MM:SS;FF) format.

- **Type**: string
- **Required**: False
- **Pattern**: `^(((\[0-1]\d)|(2[0-3]))(:[0-5]\d){2}(:;\[0-5]\d))$`

opacity
Use Opacity (Opacity) to specify how much of the underlying video shows through the inserted image. 0 is transparent and 100 is fully opaque. Default is 50.

- **Type**: integer
Properties

- **Required**: True
- **Minimum**: 0
- **Maximum**: 100

**layer**

Use Layer (Layer) to specify how overlapping inserted images appear. Images with higher values of layer appear on top of images with lower values of layer.

- **Type**: integer
- **Required**: True
- **Minimum**: 0
- **Maximum**: 99

**height**

Specify the Height (Height) of the inserted image. Use a value that is less than or equal to the video resolution height. Leave this setting blank to use the native height of the image.

- **Type**: integer
- **Required**: False
- **Minimum**: -2147483648
- **Maximum**: 2147483647

**imageInserterInput**

Use Image location (imageInserterInput) to specify the Amazon S3 location of the image to be inserted into the output. Use a 32 bit BMP, PNG, or TGA file that fits inside the video frame.

- **Type**: string
- **Required**: True
- **Pattern**: ^s3:\/[\/]\.(bmp|BMP|png|PNG|tga|TGA)$

**Job**

**settings**

- **Type**: JobSettings (p. 460)
- **Required**: True

**role**

The IAM role you use for creating this job. For details about permissions, see the User Guide topic at the User Guide at http://docs.aws.amazon.com/mediaconvert/latest/ug/iam-role.html

- **Type**: string
- **Required**: True

**outputGroupDetails**

List of output group details

- **Type**: Array of type OutputGroupDetail (p. 490)
- **Required**: False
jobTemplate

The job template that the job is created from, if it is created from a job template.

  Type: string
  Required: False

timing

  Type: Timing (p. 500)
  Required: False

errorMessage

Error message of Job

  Type: string
  Required: False

errorCode

Error code for the job

  Type: integer
  Required: False
  Format: int32

createdAt

The time, in Unix epoch format in seconds, when the job got created.

  Type: string
  Required: False
  Format: date-time

id

A portion of the job's ARN, unique within your AWS Elemental MediaConvert resources

  Type: string
  Required: False

arn

An identifier for this resource that is unique within all of AWS.

  Type: string
  Required: False

userMetadata

User-defined metadata that you want to associate with an MediaConvert job. You specify metadata in key/value pairs.
Type: object
Required: False

queue

Optional. When you create a job, you can specify a queue to send it to. If you don’t specify, the job will go to the default queue. For more about queues, see the User Guide topic at http://docs.aws.amazon.com/mediaconvert/latest/ug/what-is.html

Type: string
Required: False

status

Type: string
Required: False

JobSettings

timecodeConfig

Contains settings used to acquire and adjust timecode information from inputs.

Type: TimecodeConfig (p. 499)
Required: False

adAvailOffset

When specified, this offset (in milliseconds) is added to the input Ad Avail PTS time.

Type: integer
Required: False
Minimum: -1000
Maximum: 1000

outputGroups

(OutputGroups) contains one group of settings for each set of outputs that share a common package type. All unpackaged files (MPEG-4, MPEG-2 TS, Quicktime, MXF, and no container) are grouped in a single output group as well. Required in (OutputGroups) is a group of settings that apply to the whole group. This required object depends on the value you set for (Type) under (OutputGroups)>(OutputGroupSettings). Type, settings object pairs are as follows. * FILE_GROUP_SETTINGS, FileGroupSettings * HLS_GROUP_SETTINGS, HLSGroupSettings * DASH_ISO_GROUP_SETTINGS, DashIsoGroupSettings * MS_SMOOTH_GROUP_SETTINGS, MsSmoothGroupSettings * CMAF_GROUP_SETTINGS, CmafGroupSettings

Type: Array of type OutputGroup (p. 490)
Required: True

nielsenConfiguration

Type: NielsenConfiguration (p. 486)
Required: False
inputs

Use Inputs (inputs) to define source file used in the transcode job. There can be multiple inputs added in a job. These inputs will be concatenated together to create the output.

_Type_: Array of type `Input (p. 453)`
_Required_: True

timedMetadataInsertion

_Type_: `TimedMetadataInsertion (p. 500)`
_Required_: False

availBlanking

Settings for ad avail blanking. Video can be blanked or overlaid with an image, and audio muted during SCTE-35 triggered ad avails.

_Type_: `AvailBlanking (p. 394)`
_Required_: False

JobStatus (enum)

A job's status can be SUBMITTED, PROGRESSING, COMPLETE, CANCELED, or ERROR.

- SUBMITTED
- PROGRESSING
- COMPLETE
- CANCELED
- ERROR

LanguageCode (enum)


- ENG
- SPA
- FRA
- DEU
- GER
- ZHO
- ARA
- HIN
- JPN
- RUS
- POR
- ITA
- URD
- VIE
- KOR
- PAN
- ABK
- AAR
- AFR
HAU
HEB
HER
HMO
HUN
ISL
IDO
IBO
IND
INA
ILE
IKU
IPK
GLE
JAV
KAL
KAN
KAU
KAS
KAZ
KIK
KIN
KIR
KOM
KON
KUA
KUR
LAO
LAT
LAV
LIM
LIN
LIT
LUB
LTZ
MKD
MLG
MSA
MAL
MLT
GLV
MRI
MAR
MAH
MON
NAU
NAV
NDE
NBL
NDO
NEP
SME
NOR
NOB
NNO
<table>
<thead>
<tr>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>OCI</td>
</tr>
<tr>
<td>OJI</td>
</tr>
<tr>
<td>ORI</td>
</tr>
<tr>
<td>ORM</td>
</tr>
<tr>
<td>OSS</td>
</tr>
<tr>
<td>PLI</td>
</tr>
<tr>
<td>FAS</td>
</tr>
<tr>
<td>POL</td>
</tr>
<tr>
<td>PUS</td>
</tr>
<tr>
<td>QUE</td>
</tr>
<tr>
<td>QAA</td>
</tr>
<tr>
<td>RON</td>
</tr>
<tr>
<td>ROH</td>
</tr>
<tr>
<td>RUN</td>
</tr>
<tr>
<td>SMO</td>
</tr>
<tr>
<td>SAG</td>
</tr>
<tr>
<td>SAN</td>
</tr>
<tr>
<td>SRD</td>
</tr>
<tr>
<td>SRB</td>
</tr>
<tr>
<td>SNA</td>
</tr>
<tr>
<td>III</td>
</tr>
<tr>
<td>SND</td>
</tr>
<tr>
<td>SIN</td>
</tr>
<tr>
<td>SLK</td>
</tr>
<tr>
<td>SLV</td>
</tr>
<tr>
<td>SOM</td>
</tr>
<tr>
<td>SOT</td>
</tr>
<tr>
<td>SUN</td>
</tr>
<tr>
<td>SWA</td>
</tr>
<tr>
<td>SSW</td>
</tr>
<tr>
<td>SWE</td>
</tr>
<tr>
<td>TGL</td>
</tr>
<tr>
<td>TAH</td>
</tr>
<tr>
<td>TGK</td>
</tr>
<tr>
<td>TAM</td>
</tr>
<tr>
<td>TAT</td>
</tr>
<tr>
<td>TEL</td>
</tr>
<tr>
<td>THA</td>
</tr>
<tr>
<td>BOD</td>
</tr>
<tr>
<td>TIR</td>
</tr>
<tr>
<td>TON</td>
</tr>
<tr>
<td>TSO</td>
</tr>
<tr>
<td>TSN</td>
</tr>
<tr>
<td>TUR</td>
</tr>
<tr>
<td>TUK</td>
</tr>
<tr>
<td>TWI</td>
</tr>
<tr>
<td>UIG</td>
</tr>
<tr>
<td>UKR</td>
</tr>
<tr>
<td>UZB</td>
</tr>
<tr>
<td>VEN</td>
</tr>
<tr>
<td>VOL</td>
</tr>
<tr>
<td>WLN</td>
</tr>
<tr>
<td>CYM</td>
</tr>
<tr>
<td>FRY</td>
</tr>
<tr>
<td>WOL</td>
</tr>
</tbody>
</table>
ListJobsRequest

nextToken
Use this string, provided with the response to a previous request, to request the next batch of jobs.

Type: string
Required: False

maxResults
Optional. Number of jobs, up to twenty, that will be returned at one time.

Type: integer
Required: False
Format: int32
Minimum: 1
Maximum: 20

queue
Provide a queue name to get back only jobs from that queue.

Type: string
Required: False

status

Type: string
Required: False

order

Type: string
Required: False

ListJobsResponse

nextToken
Use this string to request the next batch of jobs.

Type: string
Required: False
jobs
List of jobs

Type: Array of type Job (p. 458)
Required: False

M2tsAudioBufferModel (enum)
Selects between the DVB and ATSC buffer models for Dolby Digital audio.

DVB
ATSC

M2tsBufferModel (enum)
Controls what buffer model to use for accurate interleaving. If set to MULTIPLEX, use multiplex buffer model. If set to NONE, this can lead to lower latency, but low-memory devices may not be able to playback the stream without interruptions.

MULTIPLEX
NONE

M2tsEbpAudioInterval (enum)
When set to VIDEO_AND_FIXED_INTERVALS, audio EBP markers will be added to partitions 3 and 4. The interval between these additional markers will be fixed, and will be slightly shorter than the video EBP marker interval. When set to VIDEO_INTERVAL, these additional markers will not be inserted. Only applicable when EBP segmentation markers are is selected (segmentationMarkers is EBP or EBP_LEGACY).

VIDEO_AND_FIXED_INTERVALS
VIDEO_INTERVAL

M2tsEbpPlacement (enum)
Selects which PIDs to place EBP markers on. They can either be placed only on the video PID, or on both the video PID and all audio PIDs. Only applicable when EBP segmentation markers are is selected (segmentationMarkers is EBP or EBP_LEGACY).

VIDEO_AND_AUDIO_PIDS
VIDEO_PID

M2tsEsRateInPes (enum)
Controls whether to include the ES Rate field in the PES header.

INCLUDE
EXCLUDE

M2tsNielsenId3 (enum)
If INSERT, Nielsen inaudible tones for media tracking will be detected in the input audio and an equivalent ID3 tag will be inserted in the output.

INSERT
NONE

**M2tsPcrControl (enum)**

When set to PCR_EVERY_PES_PACKET, a Program Clock Reference value is inserted for every Packetized Elementary Stream (PES) header. This is effective only when the PCR PID is the same as the video or audio elementary stream.

- PCR_EVERY_PES_PACKET
- CONFIGURED_PCR_PERIOD

**M2tsRateMode (enum)**

When set to CBR, inserts null packets into transport stream to fill specified bitrate. When set to VBR, the bitrate setting acts as the maximum bitrate, but the output will not be padded up to that bitrate.

- VBR
- CBR

**M2tsScte35Source (enum)**

Enables SCTE-35 passthrough (scte35Source) to pass any SCTE-35 signals from input to output.

- PASSTHROUGH
- NONE

**M2tsSegmentationMarkers (enum)**

Inserts segmentation markers at each segmentation_time period. rai_segstart sets the Random Access Indicator bit in the adaptation field. rai_adapt sets the RAI bit and adds the current timecode in the private data bytes. psi_segstart inserts PAT and PMT tables at the start of segments. ebp adds Encoder Boundary Point information to the adaptation field as per OpenCable specification OC-SP-EBP-I01-130118. ebp_legacy adds Encoder Boundary Point information to the adaptation field using a legacy proprietary format.

- NONE
- RAI_SEGSTART
- RAI_ADAPT
- PSI_SEGSTART
- EBP
- EBP_LEGACY

**M2tsSegmentationStyle (enum)**

The segmentation style parameter controls how segmentation markers are inserted into the transport stream. With avails, it is possible that segments may be truncated, which can influence where future segmentation markers are inserted. When a segmentation style of "reset_cadence" is selected and a segment is truncated due to an avail, we will reset the segmentation cadence. This means the subsequent segment will have a duration of of $segmentation_time seconds. When a segmentation style of "maintain_cadence" is selected and a segment is truncated due to an avail, we will not reset the segmentation cadence. This means the subsequent segment will likely be truncated as well. However, all segments after that will have a duration of $segmentation_time seconds. Note that EBP lookahead is a slight exception to this rule.

- MAINTAIN_CADENCE
Properties

RESET_CADENCE

**M2tsSettings**

**dvbTeletextPid**
Packet Identifier (PID) for input source DVB Teletext data to this output.

 Type: integer  
 Required: False  
 Minimum: 32  
 Maximum: 8182

**nielsenId3**

 Type: string  
 Required: False

**bitrate**

The output bitrate of the transport stream in bits per second. Setting to 0 lets the muxer automatically determine the appropriate bitrate. Other common values are 3750000, 7500000, and 15000000.

 Type: integer  
 Required: False  
 Minimum: 0  
 Maximum: 2147483647

**segmentationTime**

The length in seconds of each segment. Required unless markers is set to _none_.

 Type: number  
 Required: False  
 Format: float  
 Minimum: 0.0

**audioPids**

Packet Identifier (PID) of the elementary audio stream(s) in the transport stream. Multiple values are accepted, and can be entered in ranges and/or by comma separation.

 Type: Array of type integer  
 Required: False

**rateMode**

 Type: string  
 Required: False

**maxPcrInterval**

Maximum time in milliseconds between Program Clock References (PCRs) inserted into the transport stream.
Properties

audioFramesPerPes
The number of audio frames to insert for each PES packet.

ebpAudioInterval

fragmentTime
The length in seconds of each fragment. Only used with EBP markers.

scte35Pid
Packet Identifier (PID) of the SCTE-35 stream in the transport stream.

pmtInterval
The number of milliseconds between instances of this table in the output transport stream.

privateMetadataPid
Packet Identifier (PID) of the private metadata stream in the transport stream.
Maximum: 8182

**segmentationStyle**
Type: string  
Required: False

**audioBufferModel**
Type: string  
Required: False

**programNumber**
The value of the program number field in the Program Map Table.
Type: integer  
Required: False  
Minimum: 0  
Maximum: 65535

**dvbNitSettings**
Type: DvbNitSettings (p. 411)  
Required: False

**timedMetadataPid**
Packet Identifier (PID) of the timed metadata stream in the transport stream.
Type: integer  
Required: False  
Minimum: 32  
Maximum: 8182

**scte35Source**
Type: string  
Required: False

**pmtPid**
Packet Identifier (PID) for the Program Map Table (PMT) in the transport stream.
Type: integer  
Required: False  
Minimum: 32  
Maximum: 8182

**bufferModel**
Type: string
Required: False

ebpPlacement

Type: string
Required: False

dvbSdtSettings

Type: DvbSdtSettings (p. 412)
Required: False

nullPacketBitrate

Value in bits per second of extra null packets to insert into the transport stream. This can be used if a downstream encryption system requires periodic null packets.

Type: number
Required: False
Format: float
Minimum: 0.0

pcrPid

Packet Identifier (PID) of the Program Clock Reference (PCR) in the transport stream. When no value is given, the encoder will assign the same value as the Video PID.

Type: integer
Required: False
Minimum: 32
Maximum: 8182

minEbpInterval

When set, enforces that Encoder Boundary Points do not come within the specified time interval of each other by looking ahead at input video. If another EBP is going to come in within the specified time interval, the current EBP is not emitted, and the segment is "stretched" to the next marker. The lookahead value does not add latency to the system. The Live Event must be configured elsewhere to create sufficient latency to make the lookahead accurate.

Type: integer
Required: False
Minimum: 0
Maximum: 10000

transportStreamId

The value of the transport stream ID field in the Program Map Table.

Type: integer
Required: False
Minimum: 0
Maximum: 65535
videoPid
Packet Identifier (PID) of the elementary video stream in the transport stream.

Type: integer
Required: False
Minimum: 32
Maximum: 8182

pcrControl
Type: string
Required: False

esRateInPes
Type: string
Required: False

segmentationMarkers
Type: string
Required: False

dvbTdtSettings
Type: DvbTdtSettings (p. 417)
Required: False

patInterval
The number of milliseconds between instances of this table in the output transport stream.

Type: integer
Required: False
Minimum: 0
Maximum: 1000

dvbSubPids
Packet Identifier (PID) for input source DVB Subtitle data to this output. Multiple values are accepted, and can be entered in ranges and/or by comma separation.

Type: Array of type integer
Required: False

M3u8NielsenId3 (enum)
If INSERT, Nielsen inaudible tones for media tracking will be detected in the input audio and an equivalent ID3 tag will be inserted in the output.

INSERT
NONE
M3u8PcrControl (enum)

When set to PCR_EVERY_PES_PACKET a Program Clock Reference value is inserted for every Packetized Elementary Stream (PES) header. This parameter is effective only when the PCR PID is the same as the video or audio elementary stream.

- PCR_EVERY_PES_PACKET
- CONFIGURED_PCR_PERIOD

M3u8Scte35Source (enum)

Enables SCTE-35 passthrough (scte35Source) to pass any SCTE-35 signals from input to output.

- PASSTHROUGH
- NONE

M3u8Settings

pmtPid

Packet Identifier (PID) for the Program Map Table (PMT) in the transport stream.

- Type: integer
- Required: False
- Minimum: 32
- Maximum: 8182

nielsenId3

- Type: string
- Required: False

pcrPid

Packet Identifier (PID) of the Program Clock Reference (PCR) in the transport stream. When no value is given, the encoder will assign the same value as the Video PID.

- Type: integer
- Required: False
- Minimum: 32
- Maximum: 8182

audioPids

Packet Identifier (PID) of the elementary audio stream(s) in the transport stream. Multiple values are accepted, and can be entered in ranges and/or by comma separation.

- Type: Array of type integer
- Required: False

audioFramesPerPes

The number of audio frames to insert for each PES packet.
Properties

Type: integer
Required: False
Minimum: 0
Maximum: 2147483647

scte35Pid
Packet Identifier (PID) of the SCTE-35 stream in the transport stream.

Type: integer
Required: False
Minimum: 32
Maximum: 8182

transportStreamId
The value of the transport stream ID field in the Program Map Table.

Type: integer
Required: False
Minimum: 0
Maximum: 65535

pcrControl

Type: string
Required: False

videoPid
Packet Identifier (PID) of the elementary video stream in the transport stream.

Type: integer
Required: False
Minimum: 32
Maximum: 8182

privateMetadataPid
Packet Identifier (PID) of the private metadata stream in the transport stream.

Type: integer
Required: False
Minimum: 32
Maximum: 8182

pmtInterval
The number of milliseconds between instances of this table in the output transport stream.

Type: integer
Required: False
Minimum: 0
Maximum: 1000

**patInterval**

The number of milliseconds between instances of this table in the output transport stream.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 1000

**programNumber**

The value of the program number field in the Program Map Table.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 65535

**timedMetadataPid**

Packet Identifier (PID) of the timed metadata stream in the transport stream.

- **Type**: integer
- **Required**: False
- **Minimum**: 32
- **Maximum**: 8182

**timedMetadata**

- **Type**: string
- **Required**: False

**scte35Source**

- **Type**: string
- **Required**: False

**MovClapAtom (enum)**

When enabled, include 'clap' atom if appropriate for the video output settings.

- **INCLUDE**
- **EXCLUDE**

**MovCslgAtom (enum)**

When enabled, file composition times will start at zero, composition times in the 'ctts' (composition time to sample) box for B-frames will be negative, and a 'cslg' (composition shift least greatest) box will be included per 14496-1 amendment 1. This improves compatibility with Apple players and tools.

- **INCLUDE**
EXCLUDE

**MovMpeg2FourCCControl (enum)**

When set to XDCAM, writes MPEG2 video streams into the QuickTime file using XDCAM fourcc codes. This increases compatibility with Apple editors and players, but may decrease compatibility with other players. Only applicable when the video codec is MPEG2.

- **XDCAM**
- **MPEG**

**MovPaddingControl (enum)**

If set to OMNEON, inserts Omneon-compatible padding

- **OMNEON**
- **NONE**

**MovReference (enum)**

A value of ‘external’ creates separate media files and the wrapper file (.mov) contains references to these media files. A value of ‘self_contained’ creates only a wrapper (.mov) file and this file contains all of the media.

- **SELF_CONTAINED**
- **EXTERNAL**

**MovSettings**

**reference**

- **Type**: string
- **Required**: False

**paddingControl**

- **Type**: string
- **Required**: False

**mpeg2FourCCControl**

- **Type**: string
- **Required**: False

**cslgAtom**

- **Type**: string
- **Required**: False

**clapAtom**

- **Type**: string
Required: False

Mp2Settings

channels
Set Channels to specify the number of channels in this output audio track. Choosing Mono in the console will give you 1 output channel; choosing Stereo will give you 2. In the API, valid values are 1 and 2.

Type: integer
Required: False
Minimum: 1
Maximum: 2

bitrate
Average bitrate in bits/second.

Type: integer
Required: False
Minimum: 32000
Maximum: 384000

sampleRate
Sample rate in hz.

Type: integer
Required: False
Minimum: 32000
Maximum: 48000

Mp4CslgAtom (enum)
When enabled, file composition times will start at zero, composition times in the 'ctts' (composition time to sample) box for B-frames will be negative, and a 'cslg' (composition shift least greatest) box will be included per 14496-1 amendment 1. This improves compatibility with Apple players and tools.

INCLUDE
EXCLUDE

Mp4FreeSpaceBox (enum)
Inserts a free-space box immediately after the moov box.

INCLUDE
EXCLUDE

Mp4MoovPlacement (enum)
If set to PROGRESSIVE_DOWNLOAD, the MOOV atom is relocated to the beginning of the archive as required for progressive downloading. Otherwise it is placed normally at the end.
PROGRESSIVE_DOWNLOAD
NORMAL

**Mp4Settings**

**mp4MajorBrand**

Overrides the "Major Brand" field in the output file. Usually not necessary to specify.

- **Type:** string
- **Required:** False

**moovPlacement**

- **Type:** string
- **Required:** False

**cslgAtom**

- **Type:** string
- **Required:** False

**freeSpaceBox**

- **Type:** string
- **Required:** False

**Mpeg2AdaptiveQuantization (enum)**

Adaptive quantization. Allows intra-frame quantizers to vary to improve visual quality.

- OFF
- LOW
- MEDIUM
- HIGH

**Mpeg2CodecLevel (enum)**

Use Level (Mpeg2CodecLevel) to set the MPEG-2 level for the video output.

- AUTO
- LOW
- MAIN
- HIGH1440
- HIGH

**Mpeg2CodecProfile (enum)**

Use Profile (Mpeg2CodecProfile) to set the MPEG-2 profile for the video output.

- MAIN
- PROFILE_422
**Mpeg2FramerateControl (enum)**

If you are using the console, use the Framerate setting to specify the framerate for this output. If you want to keep the same framerate as the input video, choose Follow source. If you want to do framerate conversion, choose a framerate from the dropdown list or choose Custom. The framerates shown in the dropdown list are decimal approximations of fractions. If you choose Custom, specify your framerate as a fraction. If you are creating your transcoding job specification as a JSON file without the console, use FramerateControl to specify which value the service uses for the framerate for this output. Choose INITIALIZE_FROM_SOURCE if you want the service to use the framerate from the input. Choose SPECIFIED if you want the service to use the framerate you specify in the settings FramerateNumerator and FramerateDenominator.

- INITIALIZE_FROM_SOURCE
- SPECIFIED

**Mpeg2FramerateConversionAlgorithm (enum)**

When set to INTERPOLATE, produces smoother motion during framerate conversion.

- DUPLICATE_DROP
- INTERPOLATE

**Mpeg2GopSizeUnits (enum)**

Indicates if the GOP Size in MPEG2 is specified in frames or seconds. If seconds the system will convert the GOP Size into a frame count at run time.

- FRAMES
- SECONDS

**Mpeg2InterlaceMode (enum)**

Use Interlace mode (InterlaceMode) to choose the scan line type for the output. * Top Field First (TOP_FIELD) and Bottom Field First (BOTTOM_FIELD) produce interlaced output with the entire output having the same field polarity (top or bottom first). * Follow, Default Top (FOLLOW_TOP_FIELD) and Follow, Default Bottom (FOLLOW_BOTTOM_FIELD) use the same field polarity as the source. Therefore, behavior depends on the input scan type. - If the source is interlaced, the output will be interlaced with the same polarity as the source (it will follow the source). The output could therefore be a mix of "top field first" and "bottom field first". - If the source is progressive, the output will be interlaced with "top field first" or "bottom field first" polarity, depending on which of the Follow options you chose.

- PROGRESSIVE
- TOP_FIELD
- BOTTOM_FIELD
- FOLLOW_TOP_FIELD
- FOLLOW_BOTTOM_FIELD

**Mpeg2IntraDcPrecision (enum)**

Use Intra DC precision (Mpeg2IntraDcPrecision) to set quantization precision for intra-block DC coefficients. If you choose the value auto, the service will automatically select the precision based on the per-frame compression ratio.

- AUTO
- INTRA_DC_PRECISION_8
Properties

INTRA_DC_PRECISION_9
INTRA_DC_PRECISION_10
INTRA_DC_PRECISION_11

Mpeg2ParControl (enum)

Using the API, enable ParFollowSource if you want the service to use the pixel aspect ratio from the input. Using the console, do this by choosing Follow source for Pixel aspect ratio.

INITIALIZE_FROM_SOURCE
SPECIFIED

Mpeg2QualityTuningLevel (enum)

Use Quality tuning level (Mpeg2QualityTuningLevel) to specify whether to use single-pass or multipass video encoding.

SINGLE_PASS
MULTI_PASS

Mpeg2RateControlMode (enum)

Use Rate control mode (Mpeg2RateControlMode) to specify whether the bitrate is variable (vbr) or constant (cbr).

VBR
CBR

Mpeg2SceneChangeDetect (enum)

Scene change detection (inserts I-frames on scene changes).

DISABLED
ENABLED

Mpeg2Settings

minIInterval

Enforces separation between repeated (cadence) I-frames and I-frames inserted by Scene Change Detection. If a scene change I-frame is within I-interval frames of a cadence I-frame, the GOP is shrunk and/or stretched to the scene change I-frame. GOP stretch requires enabling lookahead as well as setting I-interval. The normal cadence resumes for the next GOP. This setting is only used when Scene Change Detect is enabled. Note: Maximum GOP stretch = GOP size + Min-I-interval - 1

Type: integer
Required: False
Minimum: 0
Maximum: 30

parNumerator

Pixel Aspect Ratio numerator.
Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

gopSizeUnits
Type: string
Required: False

hrdBufferSize
Size of buffer (HRD buffer model) in bits. For example, enter five megabits as 5000000.
Type: integer
Required: False
Minimum: 0
Maximum: 47185920

qualityTuningLevel
Type: string
Required: False

maxBitrate
Maximum bitrate in bits/second. For example, enter five megabits per second as 5000000.
Type: integer
Required: False
Minimum: 1000
Maximum: 300000000

bitrate
Average bitrate in bits/second. Required for VBR and CBR. For MS Smooth outputs, bitrates must be unique when rounded down to the nearest multiple of 1000.
Type: integer
Required: False
Minimum: 1000
Maximum: 288000000

spatialAdaptiveQuantization
Type: string
Required: False

slowPal
Type: string
Required: False
intraDcPrecision
   Type: string
   Required: False

codecProfile
   Type: string
   Required: False

softness
Softness. Selects quantizer matrix, larger values reduce high-frequency content in the encoded image.
   Type: integer
   Required: False
   Minimum: 0
   Maximum: 128

framerateControl
   Type: string
   Required: False

telecine
   Type: string
   Required: False

codecLevel
   Type: string
   Required: False

framerateConversionAlgorithm
   Type: string
   Required: False

temporalAdaptiveQuantization
   Type: string
   Required: False

hrdBufferInitialFillPercentage
Percentage of the buffer that should initially be filled (HRD buffer model).
   Type: integer
   Required: False
   Minimum: 0
   Maximum: 100
framerateNumerator
Framerate numerator - framerate is a fraction, e.g. 24000 / 1001 = 23.976 fps.

- **Type**: integer
- **Required**: False
- **Minimum**: 24
- **Maximum**: 60000

gopClosedCadence
Frequency of closed GOPs. In streaming applications, it is recommended that this be set to 1 so a decoder joining mid-stream will receive an IDR frame as quickly as possible. Setting this value to 0 will break output segmenting.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 2147483647

numberBFramesBetweenReferenceFrames
Number of B-frames between reference frames.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 7

framerateDenominator
Framerate denominator.

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 1001

adaptiveQuantization

- **Type**: string
- **Required**: False

interlaceMode

- **Type**: string
- **Required**: False

gopSize
GOP Length (keyframe interval) in frames or seconds. Must be greater than zero.

- **Type**: number
- **Required**: False
**Format**: float
**Minimum**: 0.0

**syntax**

**Type**: string
**Required**: False

**parDenominator**

Pixel Aspect Ratio denominator.

**Type**: integer
**Required**: False
**Minimum**: 1
**Maximum**: 2147483647

**sceneChangeDetect**

**Type**: string
**Required**: False

**parControl**

**Type**: string
**Required**: False

**rateControlMode**

**Type**: string
**Required**: False

**Mpeg2SlowPal (enum)**

Enables Slow PAL rate conversion. 23.976fps and 24fps input is relabeled as 25fps, and audio is sped up correspondingly.

- DISABLED
- ENABLED

**Mpeg2SpatialAdaptiveQuantization (enum)**

Adjust quantization within each frame based on spatial variation of content complexity.

- DISABLED
- ENABLED

**Mpeg2Syntax (enum)**

Produces a Type D-10 compatible bitstream (SMPTE 356M-2001).

- DEFAULT
- D_10
Mpeg2Telecine (enum)

Only use Telecine (Mpeg2Telecine) when you set Framerate (Framerate) to 29.970. Set Telecine (Mpeg2Telecine) to Hard (hard) to produce a 29.97i output from a 23.976 input. Set it to Soft (soft) to produce 23.976 output and leave conversion to the player.

- NONE
- SOFT
- HARD

Mpeg2TemporalAdaptiveQuantization (enum)

Adjust quantization within each frame based on temporal variation of content complexity.

- DISABLED
- ENABLED

MsSmoothAudioDeduplication (enum)

COMBINE_DUPLICATE_STREAMS combines identical audio encoding settings across a Microsoft Smooth output group into a single audio stream.

- COMBINE_DUPLICATE_STREAMS
- NONE

MsSmoothEncryptionSettings

spekeKeyProvider

Type: SpekeKeyProvider (p. 496)
Required: True

MsSmoothGroupSettings

fragmentLength

Use Fragment length (FragmentLength) to specify the mp4 fragment sizes in seconds. Fragment length must be compatible with GOP size and framerate.

Type: integer
Required: True
Minimum: 1
Maximum: 2147483647

encryption

Type: MsSmoothEncryptionSettings (p. 485)
Required: False

audioDeduplication

Type: string
Required: False
manifestEncoding

Type: string
Required: False

destination

Use Destination (Destination) to specify the S3 output location and the output filename base.
Destination accepts format identifiers. If you do not specify the base filename in the URI, the service will
use the filename of the input file. If your job has multiple inputs, the service uses the filename of the first
input file.

Type: string
Required: False
Pattern: ^s3://

MsSmoothManifestEncoding (enum)

Use Manifest encoding (MsSmoothManifestEncoding) to specify the encoding format for the server and
client manifest. Valid options are utf8 and utf16.

UTF8
UTF16

NielsenConfiguration

distributorId

Use Distributor ID (DistributorID) to specify the distributor ID that is assigned to your organization by
Neilsen.

Type: string
Required: False

breakoutCode

Use Nielsen Configuration (NielsenConfiguration) to set the Nielsen measurement system breakout code.
Supported values are 0, 3, 7, and 9.

Type: integer
Required: False
Minimum: 0
Maximum: 9

NoiseReducer

filter

Type: string
Required: True

filterSettings

Type: NoiseReducerFilterSettings (p. 487)
**Required**: False

**spatialFilterSettings**

**Type**: NoiseReducerSpatialFilterSettings (p. 487)

**Required**: False

**NoiseReducerFilter (enum)**

Use Noise reducer filter (NoiseReducerFilter) to select one of the following spatial image filtering functions. To use this setting, you must also enable Noise reducer (NoiseReducer). * Bilateral is an edge preserving noise reduction filter. * Mean (softest), Gaussian, Lanczos, and Sharpen (sharpest) are convolution filters. * Conserve is a min/max noise reduction filter. * Spatial is a frequency-domain filter based on JND principles.

- BILATERAL
- MEAN
- GAUSSIAN
- LANCZOS
- SHARPEN
- CONSERVE
- SPATIAL

**NoiseReducerFilterSettings**

**strength**

Relative strength of noise reducing filter. Higher values produce stronger filtering.

**Type**: integer

**Required**: False

**Minimum**: 0

**Maximum**: 3

**NoiseReducerSpatialFilterSettings**

**strength**

Relative strength of noise reducing filter. Higher values produce stronger filtering.

**Type**: integer

**Required**: False

**Minimum**: 0

**Maximum**: 16

**postFilterSharpenStrength**

Specify strength of post noise reduction sharpening filter, with 0 disabling the filter and 3 enabling it at maximum strength.

**Type**: integer

**Required**: False

**Minimum**: 0
Maximum: 3

speed
The speed of the filter, from -2 (lower speed) to 3 (higher speed), with 0 being the nominal value.

Type: integer
Required: False
Minimum: -2
Maximum: 3

Order (enum)
When you request lists of resources, you can optionally specify whether they are sorted in ASCENDING or DESCENDING order. Default varies by resource.

ASCENDING
DESCENDING

Output

extension
Use Extension (Extension) to specify the file extension for outputs in File output groups. If you do not specify a value, the service will use default extensions by container type as follows * MPEG-2 transport stream, m2ts * Quicktime, mov * MXF container, mxf * MPEG-4 container, mp4 * No Container, the service will use codec extensions (e.g. AAC, H265, H265, AC3)

Type: string
Required: False

videoDescription
(VideoDescription) contains a group of video encoding settings. The specific video settings depend on the video codec you choose when you specify a value for Video codec (codec). Include one instance of (VideoDescription) per output.

Type: VideoDescription (p. 502)
Required: False

audioDescriptions
(AudioDescriptions) contains groups of audio encoding settings organized by audio codec. Include one instance of (AudioDescriptions) per output. (AudioDescriptions) can contain multiple groups of encoding settings.

Type: Array of type AudioDescription (p. 391)
Required: False

containerSettings

Type: ContainerSettings (p. 407)
Required: False
**preset**

Use Preset (Preset) to specify a preset for your transcoding settings. Provide the system or custom preset name. You can specify either Preset (Preset) or Container settings (ContainerSettings), but not both.

*Type:* string  
*Required:* False

**outputSettings**

*Type:* OutputSettings (p. 491)  
*Required:* False

**captionDescriptions**

(CaptionDescriptions) contains groups of captions settings. For each output that has captions, include one instance of (CaptionDescriptions). (CaptionDescriptions) can contain multiple groups of captions settings.

*Type:* Array of type CaptionDescription (p. 398)  
*Required:* False

**nameModifier**

Use Name modifier (NameModifier) to have the service add a string to the end of each output filename. You specify the base filename as part of your destination URI. When you create multiple outputs in the same output group, Name modifier (NameModifier) is required. Name modifier also accepts format identifiers. For DASH ISO outputs, if you use the format identifiers $Number$ or $Time$ in one output, you must use them in the same way in all outputs of the output group.

*Type:* string  
*Required:* False

**OutputChannelMapping**

**inputChannels**

List of input channels

*Type:* Array of type integer  
*Required:* True

**OutputDetail**

**durationInMs**

Duration in milliseconds

*Type:* integer  
*Required:* False

**videoDetails**

*Type:* VideoDetail (p. 504)
**OutputGroup**

**outputs**

This object holds groups of encoding settings, one group of settings per output.

- **Type:** Array of type Output (p. 488)
- **Required:** True

**outputGroupSettings**

- **Type:** OutputGroupSettings (p. 490)
- **Required:** True

**name**

Name of the output group

- **Type:** string
- **Required:** False

**customName**

Use Custom Group Name (CustomName) to specify a name for the output group. This value is displayed on the console and can make your job settings JSON more human-readable. It does not affect your outputs. Use up to twelve characters that are either letters, numbers, spaces, or underscores.

- **Type:** string
- **Required:** False

**OutputGroupDetail**

**outputDetails**

Details about the output

- **Type:** Array of type OutputDetail (p. 489)
- **Required:** False

**OutputGroupSettings**

**dashIsoGroupSettings**

- **Type:** DashIsoGroupSettings (p. 409)
- **Required:** False

**fileGroupSettings**

- **Type:** FileGroupSettings (p. 423)
**Properties**

**Required:** False

**msSmoothGroupSettings**
- **Type:** MsSmoothGroupSettings (p. 485)
- **Required:** False

**cmafGroupSettings**
- **Type:** CmafGroupSettings (p. 402)
- **Required:** False

**type**
- **Type:** string
- **Required:** True

**hlsGroupSettings**
- **Type:** HlsGroupSettings (p. 447)
- **Required:** False

**OutputGroupType (enum)**
Type of output group (File group, Apple HLS, DASH ISO, Microsoft Smooth Streaming, CMAF)

- HLS_GROUP_SETTINGS
- DASH_ISO_GROUP_SETTINGS
- FILE_GROUP_SETTINGS
- MS_SMOOTH_GROUP_SETTINGS
- CMAF_GROUP_SETTINGS

**OutputSdt (enum)**
Selects method of inserting SDT information into output stream. "Follow input SDT" copies SDT information from input stream to output stream. "Follow input SDT if present" copies SDT information from input stream to output stream if SDT information is present in the input, otherwise it will fall back on the user-defined values. Enter "SDT Manually" means user will enter the SDT information. "No SDT" means output stream will not contain SDT information.

- SDT_FOLLOW
- SDT_FOLLOW_IF_PRESENT
- SDT_MANUAL
- SDT_NONE

**OutputSettings**

**hlsSettings**
- **Type:** HlsSettings (p. 451)
- **Required:** False
Properties

**ProresCodecProfile (enum)**

Use Profile (ProResCodecProfile) to specify the type of Apple ProRes codec to use for this output.

- APPLE_PRORES_422
- APPLE_PRORES_422_HQ
- APPLE_PRORES_422_LT
- APPLE_PRORES_422_PROXY

**ProresFramerateControl (enum)**

If you are using the console, use the Framerate setting to specify the framerate for this output. If you want to keep the same framerate as the input video, choose Follow source. If you want to do framerate conversion, choose a framerate from the dropdown list or choose Custom. The framerates shown in the dropdown list are decimal approximations of fractions. If you choose Custom, specify your framerate as a fraction. If you are creating your transcoding job specification as a JSON file without the console, use FramerateControl to specify which value the service uses for the framerate for this output. Choose INITIALIZE_FROM_SOURCE if you want the service to use the framerate from the input. Choose SPECIFIED if you want the service to use the framerate you specify in the settings FramerateNumerator and FramerateDenominator.

- INITIALIZE_FROM_SOURCE
- SPECIFIED

**ProresFramerateConversionAlgorithm (enum)**

When set to INTERPOLATE, produces smoother motion during framerate conversion.

- DUPLICATE_DROP
- INTERPOLATE

**ProresInterlaceMode (enum)**

Use Interlace mode (InterlaceMode) to choose the scan line type for the output. * Top Field First (TOP_FIELD) and Bottom Field First (BOTTOM_FIELD) produce interlaced output with the entire output having the same field polarity (top or bottom first). * Follow, Default Top (FOLLOW_TOP_FIELD) and Follow, Default Bottom (FOLLOW_BOTTOM_FIELD) use the same field polarity as the source. Therefore, behavior depends on the input scan type. - If the source is interlaced, the output will be interlaced with the same polarity as the source (it will follow the source). The output could therefore be a mix of “top field first” and “bottom field first”. - If the source is progressive, the output will be interlaced with “top field first” or “bottom field first” polarity, depending on which of the Follow options you chose.

- PROGRESSIVE
- TOP_FIELD
- BOTTOM_FIELD
- FOLLOW_TOP_FIELD
- FOLLOW_BOTTOM_FIELD

**ProresParControl (enum)**

Use (ProresParControl) to specify how the service determines the pixel aspect ratio. Set to Follow source (INITIALIZE_FROM_SOURCE) to use the pixel aspect ratio from the input. To specify a different pixel aspect ratio: Using the console, choose it from the dropdown menu. Using the API, set ProresParControl to (SPECIFIED) and provide for (ParNumerator) and (ParDenominator).
INITIALIZE_FROM_SOURCE
SPECIFIED

**ProresSettings**

**framerateDenominator**

Framerate denominator.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 2147483647

**slowPal**

- **Type:** string
- **Required:** False

**framerateControl**

- **Type:** string
- **Required:** False

**telecine**

- **Type:** string
- **Required:** False

**framerateConversionAlgorithm**

- **Type:** string
- **Required:** False

**interlaceMode**

- **Type:** string
- **Required:** False

**parNumerator**

Pixel Aspect Ratio numerator.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 2147483647

**codecProfile**

- **Type:** string
Required: False

parDenominator

Pixel Aspect Ratio denominator.

Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

parControl

Type: string
Required: False

framerateNumerator

When you use the API for transcoding jobs that use framerate conversion, specify the framerate as a fraction. For example, 24000 / 1001 = 23.976 fps. Use FramerateNumerator to specify the numerator of this fraction. In this example, use 24000 for the value of FramerateNumerator.

Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

ProresSlowPal (enum)

Enables Slow PAL rate conversion. 23.976fps and 24fps input is relabeled as 25fps, and audio is sped up correspondingly.

DISABLED
ENABLED

ProresTelecine (enum)

Only use Telecine (ProresTelecine) when you set Framerate (Framerate) to 29.970. Set Telecine (ProresTelecine) to Hard (hard) to produce a 29.97i output from a 23.976 input. Set it to Soft (soft) to produce 23.976 output and leave conversion to the player.

NONE
HARD

Rectangle

width

Width of rectangle in pixels.

Type: integer
Required: True
Minimum: -2147483648
Properties

**Maximum**: 2147483647

**x**
The distance, in pixels, between the rectangle and the left edge of the video frame.

- **Type**: integer
- **Required**: True
- **Minimum**: -2147483648
- **Maximum**: 2147483647

**y**
The distance, in pixels, between the rectangle and the top edge of the video frame.

- **Type**: integer
- **Required**: True
- **Minimum**: -2147483648
- **Maximum**: 2147483647

**height**
Height of rectangle in pixels.

- **Type**: integer
- **Required**: True
- **Minimum**: -2147483648
- **Maximum**: 2147483647

**RemixSettings**

**channelsOut**
Specify the number of channels in this output after remixing. Valid values: 1, 2, 4, 6, 8

- **Type**: integer
- **Required**: True
- **Minimum**: 1
- **Maximum**: 8

**channelMapping**

- **Type**: ChannelMapping (p. 401)
- **Required**: True

**channelsIn**
Specify the number of audio channels from your input that you want to use in your output. With remixing, you might combine or split the data in these channels, so the number of channels in your final output might be different.

- **Type**: integer
- **Required**: True
Properties

**Minimum**: 1  
**Maximum**: 16

**RespondToAfd (enum)**

Use Respond to AFD (RespondToAfd) to specify how the service changes the video itself in response to AFD values in the input. * Choose Respond to clip the input video frame according to the AFD value, input display aspect ratio, and output display aspect ratio. * Choose Passthrough to include the input AFD values. Do not choose this when AfdSignaling is set to (NONE). A preferred implementation of this workflow is to set RespondToAfd to (NONE) and set AfdSignaling to (AUTO). * Choose None to remove all input AFD values from this output.

- NONE  
- RESPOND  
- PASSTHROUGH

**ScalingBehavior (enum)**

Applies only if your input aspect ratio is different from your output aspect ratio. Enable Stretch to output (StretchToOutput) to have the service stretch your video image to fit. Leave this setting disabled to allow the service to letterbox your video instead. This setting overrides any positioning value you specify elsewhere in the job.

- DEFAULT  
- STRETCH_TO_OUTPUT

**SccDestinationFramerate (enum)**

Set Framerate (SccDestinationFramerate) to make sure that the captions and the video are synchronized in the output. Specify a framerate that matches the framerate of the associated video. If the video framerate is 29.97, choose 29.97 dropframe (FRAMERATE_29_97_DROPFRAME) only if the video has video_insertion=true and drop_frame_timecode=true; otherwise, choose 29.97 non-dropframe (FRAMERATE_29_97_NON_DROPFRAME).

- FRAMERATE_23_97  
- FRAMERATE_24  
- FRAMERATE_29_97_DROPFRAME  
- FRAMERATE_29_97_NON_DROPFRAME

**SccDestinationSettings**

**framerate**

- **Type**: string  
- **Required**: False

**SpekeKeyProvider**

**resourceld**

The SPEKE-compliant server uses Resource ID (Resourceld) to identify content.

- **Type**: string
**Required**: True  
**Pattern**: `^[\w-]+$`

**systemIds**

Relates to SPEKE implementation. DRM system identifiers. DASH output groups support a max of two system ids. Other group types support one system id.

**Type**: Array of type string  
**Required**: True

**url**

Use URL (Url) to specify the SPEKE-compliant server that will provide keys for content.

**Type**: string  
**Required**: True  
**Format**: uri  
**Pattern**: `^https:/\/\/`

**StaticKeyProvider**

**keyFormatVersions**

Relates to DRM implementation. Either a single positive integer version value or a slash delimited list of version values (1/2/3).

**Type**: string  
**Required**: False  
**Pattern**: `^\d+(\d+)*$`

**keyFormat**

Relates to DRM implementation. Sets the value of the KEYFORMAT attribute. Must be 'identity' or a reverse DNS string. May be omitted to indicate an implicit value of 'identity'.

**Type**: string  
**Required**: False  
**Pattern**: `^(identity|\[A-Za-z\]{2,6}(\.[A-Za-z0-9-]{1,63})+)$`

**staticKeyValue**

Relates to DRM implementation. Use a 32-character hexadecimal string to specify Key Value (StaticKeyValue).

**Type**: string  
**Required**: True  
**Pattern**: `^[A-Za-z0-9]{32}$`

**url**

Relates to DRM implementation. The location of the license server used for protecting content.

**Type**: string
TeletextDestinationSettings

pageNumber

Set pageNumber to the Teletext page number for the destination captions for this output. This value must be a three-digit hexadecimal string; strings ending in -FF are invalid. If you are passing through the entire set of Teletext data, do not use this field.

Type: string
Required: False
Pattern: ^[1-8][0-9a-fA-F][0-9a-eA-E]$

TeletextSourceSettings

pageNumber

Use Page Number (PageNumber) to specify the three-digit hexadecimal page number that will be used for Teletext captions. Do not use this setting if you are passing through teletext from the input source to output.

Type: string
Required: False
Pattern: ^[1-8][0-9a-fA-F][0-9a-eA-E]$

TimecodeBurnin

prefix

Use Prefix (Prefix) to place ASCII characters before any burned-in timecode. For example, a prefix of "EZ-" will result in the timecode "EZ-00:00:00:00". Provide either the characters themselves or the ASCII code equivalents. The supported range of characters is 0x20 through 0x7e. This includes letters, numbers, and all special characters represented on a standard English keyboard.

Type: string
Required: False
Pattern: ^[- ~]+$

fontSize

Use Font Size (FontSize) to set the font size of any burned-in timecode. Valid values are 10, 16, 32, 48.

Type: integer
Required: False
Minimum: 10
Maximum: 48

position

Type: string
Required: False
**TimecodeBurninPosition (enum)**

Use Position (Position) under Timecode burn-in (TimecodeBurnIn) to specify the location the burned-in timecode on output video.

- TOP_CENTER
- TOP_LEFT
- TOP_RIGHT
- MIDDLE_LEFT
- MIDDLE_CENTER
- MIDDLE_RIGHT
- BOTTOM_LEFT
- BOTTOM_CENTER
- BOTTOM_RIGHT

**TimecodeConfig**

**timestampOffset**

Only applies to outputs that support program-date-time stamp. Use Timestamp offset (TimestampOffset) to overwrite the timecode date without affecting the time and frame number. Provide the new date as a string in the format "yyyy-mm-dd". To use Time stamp offset, you must also enable Insert program-date-time (InsertProgramDateTime) in the output settings. For example, if the date part of your timecodes is 2002-1-25 and you want to change it to one year later, set Timestamp offset (TimestampOffset) to 2003-1-25.

- **Type:** string
- **Required:** False
- **Pattern:** `^([0-9]{4})-(0[1-9]|1[0-2])-(0[1-9]|12)(0-9|301)$`

**anchor**

If you use an editing platform that relies on an anchor timecode, use Anchor Timecode (Anchor) to specify a timecode that will match the input video frame to the output video frame. Use 24-hour format with frame number, (HH:MM:SS:FF) or (HH:MM:SS;FF). This setting ignores framerate conversion. System behavior for Anchor Timecode varies depending on your setting for Source (TimecodeSource).

- If Source (TimecodeSource) is set to Specified Start (SPECIFIEDSTART), the first input frame is the specified value in Start Timecode (Start). Anchor Timecode (Anchor) and Start Timecode (Start) are used to calculate output timecode. * If Source (TimecodeSource) is set to Start at 0 (ZEROBASED) the first frame is 00:00:00:00. * If Source (TimecodeSource) is set to Embedded (EMBEDDED), the first frame is the timecode value on the first input frame of the input.

- **Type:** string
- **Required:** False
- **Format:** timecode
- **Pattern:** `^([01][0-9]|2[0-4]):[0-5][0-9]:[0-5][0-9]:[0-9]$`

**start**

Only use when you set Source (TimecodeSource) to Specified start (SPECIFIEDSTART). Use Start timecode (Start) to specify the timecode for the initial frame. Use 24-hour format with frame number, (HH:MM:SS:FF) or (HH:MM:SS;FF).

- **Type:** string
- **Required:** False
Format: timecode
Pattern: ^([01][0-9]|2[0-4]):[0-5][0-9]:[0-5][0-9]:[;][0-9]{2}$

source
Type: string
Required: False

TimecodeSource (enum)
Use Source (TimecodeSource) to set how timecodes are handled within this job. To make sure that your video, audio, captions, and markers are synchronized and that time-based features, such as image inserter, work correctly, choose the Timecode source option that matches your assets. All timecodes are in a 24-hour format with frame number (HH:MM:SS:FF).

- Embedded (EMBEDDED) - Use the timecode that is in the input video. If no embedded timecode is in the source, the service will use Start at 0 (ZEROBASED) instead.
- Start at 0 (ZEROBASED) - Set the timecode of the initial frame to 00:00:00:00.
- Specified Start (SPECIFIEDSTART) - Set the timecode of the initial frame to a value other than zero. You use Start timecode (Start) to provide this value.

  EMBEDDED
  ZEROBASED
  SPECIFIEDSTART

TimedMetadata (enum)
Applies only to HLS outputs. Use this setting to specify whether the service inserts the ID3 timed metadata from the input in this output.

  PASSTHROUGH
  NONE

TimedMetadataInsertion
id3Insertions
Id3Insertions contains the array of Id3Insertion instances.

  Type: Array of type Id3Insertion (p. 452)
  Required: True

Timing
finishTime
The time, in Unix epoch format, that the transcoding job finished

  Type: string
  Required: False
  Format: date-time

submitTime
The time, in Unix epoch format, that you submitted the job.
**Properties**

**Type**
- **Type**: string
- **Required**: False
- **Format**: date-time

**startTime**

The time, in Unix epoch format, that transcoding for the job began.

- **Type**: string
- **Required**: False
- **Format**: date-time

**TtmlDestinationSettings**

**stylePassthrough**

- **Type**: string
- **Required**: False

**TtmlStylePassthrough (enum)**

Pass through style and position information from a TTML-like input source (TTML, SMPTE-TT, CFF-TT) to the CFF-TT output or TTML output.

- **ENABLED**
- **DISABLED**

**VideoCodec (enum)**

Type of video codec

- **FRAME_CAPTURE**
- **H_264**
- **H_265**
- **MPEG2**
- **PRORES**

**VideoCodecSettings**

**h265Settings**

- **Type**: H265Settings (p. 436)
- **Required**: False

**codec**

- **Type**: string
- **Required**: True

**proresSettings**

- **Type**: ProresSettings (p. 493)
Required: False

mpeg2Settings

Type: Mpeg2Settings (p. 480)
Required: False

frameCaptureSettings

Type: FrameCaptureSettings (p. 424)
Required: False

h264Settings

Type: H264Settings (p. 428)
Required: False

VideoDescription

fixedAfd

Applies only if you set AFD Signaling(AfdSignaling) to Fixed (FIXED). Use Fixed (FixedAfd) to specify a four-bit AFD value which the service will write on all frames of this video output.

Type: integer
Required: False
Minimum: 0
Maximum: 15

scalingBehavior

Type: string
Required: False

respondToAfd

Type: string
Required: False

codecSettings

Type: VideoCodecSettings (p. 501)
Required: True

afdSignaling

Type: string
Required: False

colorMetadata

Type: string
Required: False

timecodeInsertion

Type: string
Required: False

width

Use Width (Width) to define the video resolution width, in pixels, for this output. If you don't provide a value here, the service will use the input width.

Type: integer
Required: False
Minimum: 32
Maximum: 4096

videoPreprocessors

Find additional transcoding features under Preprocessors (VideoPreprocessors). Enable the features at each output individually. These features are disabled by default.

Type: VideoPreprocessor (p. 504)
Required: False

antiAlias

Type: string
Required: False

position

Use Position (Position) to point to a rectangle object to define your position. This setting overrides any other aspect ratio.

Type: Rectangle (p. 494)
Required: False

sharpness

Use Sharpness (Sharpness) setting to specify the strength of anti-aliasing. This setting changes the width of the anti-alias filter kernel used for scaling. Sharpness only applies if your output resolution is different from your input resolution, and if you set Anti-alias (AntiAlias) to ENABLED. 0 is the softest setting, 100 the sharpest, and 50 recommended for most content.

Type: integer
Required: False
Minimum: 0
Maximum: 100

crop

Applies only if your input aspect ratio is different from your output aspect ratio. Use Input cropping rectangle (Crop) to specify the video area the service will include in the output. This will crop the input
source, causing video pixels to be removed on encode. Do not use this setting if you have enabled Stretch to output (stretchToOutput) in your output settings.

- **Type**: Rectangle (p. 494)
- **Required**: False

**dropFrameTimecode**

- **Type**: string
- **Required**: False

**height**

Use the Height (Height) setting to define the video resolution height for this output. Specify in pixels. If you don’t provide a value here, the service will use the input height.

- **Type**: integer
- **Required**: False
- **Minimum**: 32
- **Maximum**: 2160

**VideoDetail**

**heightInPx**

Height in pixels for the output

- **Type**: integer
- **Required**: False

**widthInPx**

Width in pixels for the output

- **Type**: integer
- **Required**: False

**VideoPreprocessor**

**noiseReducer**

Enable the Noise reducer (NoiseReducer) feature to remove noise from your video output if necessary. Enable or disable this feature for each output individually. This setting is disabled by default.

- **Type**: NoiseReducer (p. 486)
- **Required**: False

**timecodeBurnin**

Timecode burn-in (TimecodeBurnIn)—Burns the output timecode and specified prefix into the output.

- **Type**: TimecodeBurnin (p. 498)
- **Required**: False
Properties

**colorCorrector**
Enable the Color corrector (ColorCorrector) feature if necessary. Enable or disable this feature for each output individually. This setting is disabled by default.

*Type: ColorCorrector (p. 405)*  
*Required: False*

**deinterlacer**
Use Deinterlacer (Deinterlacer) to produce smoother motion and a clearer picture.

*Type: Deinterlacer (p. 411)*  
*Required: False*

**imageInserter**
Enable the Image inserter (ImageInserter) feature to include a graphic overlay on your video. Enable or disable this feature for each output individually. This setting is disabled by default.

*Type: ImageInserter (p. 453)*  
*Required: False*

**VideoSelector**

**colorSpace**

*Type: string*
*Required: False*

**hdr10Metadata**

*Type: Hdr10Metadata (p. 442)*  
*Required: False*

**programNumber**
Selects a specific program from within a multi-program transport stream. Note that Quad 4K is not currently supported.

*Type: integer*
*Required: False*
*Minimum: -2147483648*
*Maximum: 2147483647*

**pid**
Use PID (Pid) to select specific video data from an input file. Specify this value as an integer; the system automatically converts it to the hexadecimal value. For example, 257 selects PID 0x101. A PID, or packet identifier, is an identifier for a set of data in an MPEG-2 transport stream container.

*Type: integer*
*Required: False*
*Minimum: 1*
*Maximum: 2147483647*
**colorSpaceUsage**

**Type:** string  
**Required:** False

**VideoTimecodeInsertion (enum)**

Applies only to H.264, H.265, MPEG2, and ProRes outputs. Only enable Timecode insertion when the input framerate is identical to the output framerate. To include timecodes in this output, set Timecode insertion (VideoTimecodeInsertion) to PIC_TIMING_SEI. To leave them out, set it to DISABLED. Default is DISABLED. When the service inserts timecodes in an output, by default, it uses any embedded timecodes from the input. If none are present, the service will set the timecode for the first output frame to zero. To change this default behavior, adjust the settings under Timecode configuration (TimecodeConfig). In the console, these settings are located under Job > Job settings > Timecode configuration. Note - Timecode source under input settings (InputTimecodeSource) does not affect the timecodes that are inserted in the output. Source under Job settings > Timecode configuration (TimecodeSource) does.

- DISABLED  
- PIC_TIMING_SEI

**WavFormat (enum)**

The service defaults to using RIFF for WAV outputs. If your output audio is likely to exceed 4 GB in file size, or if you otherwise need the extended support of the RF64 format, set your output WAV file format to RF64.

- RIFF  
- RF64

**WavSettings**

**channels**

Set Channels to specify the number of channels in this output audio track. With WAV, valid values 1, 2, 4, and 8. In the console, these values are Mono, Stereo, 4-Channel, and 8-Channel, respectively.

**Type:** integer  
**Required:** False  
**Minimum:** 1  
**Maximum:** 8

**bitDepth**

Specify Bit depth (BitDepth), in bits per sample, to choose the encoding quality for this audio track.

**Type:** integer  
**Required:** False  
**Minimum:** 16  
**Maximum:** 24

**format**

**Type:** string
Jobs id

URI

/2017-08-29/jobs/ \(id\)

HTTP Methods

GET

Operation ID: GetJob

Retrieve the JSON for a specific completed transcoding job.

Path Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>String</td>
<td>True</td>
<td></td>
</tr>
</tbody>
</table>

Responses

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>GetJobResponse (p. 508)</td>
<td>200 response</td>
</tr>
<tr>
<td>400</td>
<td>ExceptionBody (p. 519)</td>
<td>BadRequestException</td>
</tr>
<tr>
<td>500</td>
<td>ExceptionBody (p. 519)</td>
<td>InternalServiceException</td>
</tr>
<tr>
<td>403</td>
<td>ExceptionBody (p. 519)</td>
<td>AccessDeniedException</td>
</tr>
<tr>
<td>404</td>
<td>ExceptionBody (p. 519)</td>
<td>ResourceNotFoundException</td>
</tr>
<tr>
<td>429</td>
<td>ExceptionBody (p. 519)</td>
<td>LimitExceededException</td>
</tr>
<tr>
<td>409</td>
<td>ExceptionBody (p. 519)</td>
<td>ResourceInUseException</td>
</tr>
</tbody>
</table>

DELETE

Operation ID: CancelJob
Permanently remove a job from a queue. Once you have canceled a job, you can't start it again. You can't delete a running job.

Path Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>String</td>
<td>True</td>
<td></td>
</tr>
</tbody>
</table>

Responses

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>ExceptionBody (p. 519)</td>
<td>BadRequestException</td>
</tr>
<tr>
<td>202</td>
<td>CancelJobResponse (p. 519)</td>
<td>202 response</td>
</tr>
<tr>
<td>500</td>
<td>ExceptionBody (p. 519)</td>
<td>InternalServiceException</td>
</tr>
<tr>
<td>403</td>
<td>ExceptionBody (p. 519)</td>
<td>AccessDeniedException</td>
</tr>
<tr>
<td>404</td>
<td>ExceptionBody (p. 519)</td>
<td>ResourceNotFoundException</td>
</tr>
<tr>
<td>429</td>
<td>ExceptionBody (p. 519)</td>
<td>LimitExceededException</td>
</tr>
<tr>
<td>409</td>
<td>ExceptionBody (p. 519)</td>
<td>ResourceInUseException</td>
</tr>
</tbody>
</table>

Schemas

Request Bodies

Example GET

```json
{
  "id (p. 559)": "string"
}
```

Example DELETE

```json
{
  "id (p. 533)": "string"
}
```

Response Bodies

Example GetJobResponse

```json
{
  "job (p. 559)": {
    "settings (p. 592)": {
      "timecodeConfig (p. 594)": {
        "timestampOffset (p. 632)": "string",
        "anchor (p. 632)": "string",
        "start (p. 633)": "string"
      }
    }
  }
}
```
"source (p. 633)" : enum,
"adAvailOffset (p. 594)" : integer,
"outputGroups (p. 594)" : [ 
  "outputs (p. 623)" : [ 
    "extension (p. 621)" : "string",
    "videoDescription (p. 621)" : { 
      "fixedAfd (p. 635)" : integer,
      "scalingBehavior (p. 635)" : enum,
      "respondToAfd (p. 635)" : enum,
      "codecSettings (p. 635)" : { 
        "h265Settings (p. 634)" : { 
          "slices (p. 570)" : integer,
          "minIInterval (p. 571)" : integer,
          "parNumerator (p. 571)" : integer,
          "flickerAdaptiveQuantization (p. 571)" : enum,
          "gopSizeUnits (p. 571)" : enum,
          "hrdBufferSize (p. 571)" : integer,
          "qualityTuningLevel (p. 571)" : enum,
          "maxBitrate (p. 571)" : integer,
          "bitrate (p. 572)" : integer,
          "spatialAdaptiveQuantization (p. 572)" : enum,
          "sampleAdaptiveOffsetFilterMode (p. 572)" : enum,
          "temporalIds (p. 572)" : enum,
          "slowPal (p. 572)" : enum,
          "writeMp4PackagingType (p. 572)" : enum,
          "codecProfile (p. 572)" : enum,
          "alternateTransferFunctionSei (p. 572)" : enum,
          "unregisteredSeiTimecode (p. 573)" : enum,
          "framerateControl (p. 573)" : enum,
          "telecine (p. 573)" : enum,
          "framerateConversionAlgorithm (p. 573)" : enum,
          "numberReferenceFrames (p. 573)" : integer,
          "temporalAdaptiveQuantization (p. 573)" : enum,
          "framerateNumerator (p. 573)" : integer,
          "framerateDenominator (p. 573)" : integer,
          "adaptiveQuantization (p. 574)" : enum,
          "gopSize (p. 574)" : number,
          "gopBReference (p. 575)" : enum,
          "parNumerator (p. 575)" : integer,
          "parDenominator (p. 575)" : integer,
          "sceneChangeDetect (p. 575)" : enum,
          "parControl (p. 575)" : enum,
          "rateControlMode (p. 575)" : enum 
        },
        "proresSettings (p. 635)" : { 
          "framerateDenominator (p. 626)" : integer,
          "slowPal (p. 626)" : enum,
          "telecine (p. 626)" : enum,
          "framerateConversionAlgorithm (p. 626)" : enum,
          "interlaceMode (p. 626)" : enum,
          "parNumerator (p. 627)" : integer,
          "codecProfile (p. 627)" : enum,
          "parDenominator (p. 627)" : integer,
          "parControl (p. 627)" : enum,
          "framerateNumerator (p. 627)" : integer 
        },
      } 
    } 
  } 
] 
] 
]
"mpeg2Settings (p. 635)": {
    "minIInterval (p. 614)": integer,
    "parNumerator (p. 614)": integer,
    "gopSizeUnits (p. 614)": enum,
    "hrdBufferSize (p. 614)": integer,
    "qualityTuningLevel (p. 614)": enum,
    "maxBitrate (p. 614)": integer,
    "bitrate (p. 615)": integer,
    "spatialAdaptiveQuantization (p. 615)": enum,
    "slowPal (p. 615)": enum,
    "intraDcPrecision (p. 615)": enum,
    "codecProfile (p. 615)": enum,
    "qualityTuningLevel (p. 614)": enum,
    "maxBitrate (p. 614)": integer,
    "bitrate (p. 615)": integer,
    "spatialAdaptiveQuantization (p. 615)": enum,
    "slowPal (p. 615)": enum,
    "intraDcPrecision (p. 615)": enum,
    "codecProfile (p. 615)": enum,
    "softness (p. 615)": integer,
    "framerateControl (p. 615)": integer,
    "telecine (p. 615)": enum,
    "codecLevel (p. 615)": enum,
    "framerateConversionAlgorithm (p. 616)": enum,
    "temporalAdaptiveQuantization (p. 616)": enum,
    "hrdBufferInitialFillPercentage (p. 616)": integer,
    "framerateNumerator (p. 616)": integer,
    "framerateDenominator (p. 616)": integer,
    "adaptiveQuantization (p. 617)": enum,
    "interlaceMode (p. 617)": enum,
    "gopSize (p. 617)": number,
    "syntax (p. 617)": enum,
    "parDenominator (p. 617)": integer,
    "sceneChangeDetect (p. 617)": enum,
    "parControl (p. 617)": enum,
    "rateControlMode (p. 617)": enum
},
"frameCaptureSettings (p. 635)": {
    "framerateDenominator (p. 558)": integer,
    "maxCaptures (p. 558)": integer,
    "framerateNumerator (p. 558)": integer,
    "quality (p. 558)": integer
},
"h264Settings (p. 635)": {
    "slices (p. 562)": integer,
    "minIInterval (p. 562)": integer,
    "parNumerator (p. 562)": integer,
    "flickerAdaptiveQuantization (p. 562)": enum,
    "gopSizeUnits (p. 563)": enum,
    "hrdBufferSize (p. 563)": integer,
    "qualityTuningLevel (p. 563)": enum,
    "maxBitrate (p. 563)": integer,
    "bitrate (p. 563)": integer,
    "spatialAdaptiveQuantization (p. 563)": enum,
    "slowPal (p. 563)": integer,
    "unregisteredSeiTimecode (p. 564)": enum,
    "softness (p. 564)": integer,
    "framerateControl (p. 564)": integer,
    "telecine (p. 564)": enum,
    "codecLevel (p. 564)": enum,
    "framerateConversionAlgorithm (p. 564)": enum,
    "numberReferenceFrames (p. 564)": integer,
    "temporalAdaptiveQuantization (p. 564)": enum,
    "repeatPps (p. 564)": enum,
    "hrdBufferInitialFillPercentage (p. 565)": integer,
    "framerateNumerator (p. 565)": integer,
    "framerateDenominator (p. 565)": integer,
    "entropyEncoding (p. 565)": enum,
"fieldEncoding (p. 566)": enum,
"adaptiveQuantization (p. 566)": enum,
"interlaceMode (p. 566)": enum,
"gopSize (p. 566)": number,
"gopBReference (p. 566)": enum,
"syntax (p. 566)": enum,
"parDenominator (p. 566)": integer,
"sceneChangeDetect (p. 566)": enum,
"parControl (p. 566)": enum,
"rateControlMode (p. 567)": enum
}
,"afdSignaling (p. 636)": enum,
"colorMetadata (p. 636)": enum,
"timecodeInsertion (p. 636)": enum,
"width (p. 636)": integer,
"videoPreprocessors (p. 636)": {
  "noiseReducer (p. 637)": {
    "filter (p. 620)": enum,
    "filterSettings (p. 620)": {
      "strength (p. 620)": integer
    },
    "spatialFilterSettings (p. 620)": {
      "strength (p. 621)": integer,
      "postFilterSharpenStrength (p. 621)": integer,
      "speed (p. 621)": integer
    }
  },
  "timecodeBurnin (p. 638)": {
    "prefix (p. 631)": "string",
    "fontSize (p. 631)": integer,
    "position (p. 632)": enum
  },
  "colorCorrector (p. 638)": {
    "saturation (p. 540)": integer,
    "brightness (p. 540)": integer,
    "hdr10Metadata (p. 541)": {
      "redPrimaryY (p. 576)": integer,
      "greenPrimaryY (p. 577)": integer,
      "whitePointX (p. 577)": integer,
      "maxLuminance (p. 577)": integer,
      "greenPrimaryX (p. 577)": integer,
      "whitePointY (p. 577)": integer,
      "redPrimaryX (p. 578)": integer,
      "bluePrimaryX (p. 578)": integer,
      "bluePrimaryY (p. 578)": integer,
      "maxFrameAverageLightLevel (p. 578)": integer,
      "maxContentLightLevel (p. 578)": integer,
      "minLuminance (p. 579)": integer
    },
    "contrast (p. 541)": integer,
    "hue (p. 541)": integer,
    "colorSpaceConversion (p. 541)": enum
  },
  "deinterlacer (p. 638)": {
    "mode (p. 545)": enum,
    "control (p. 545)": enum,
    "algorithm (p. 545)": enum
  },
  "imageInserter (p. 638)": {
    "insertableImages (p. 587)": [
      {
        "duration (p. 590)": integer,
        "fadeOut (p. 590)": integer,
        "imageY (p. 591)": integer,
        "fadeIn (p. 591)": integer,
"imageX (p. 591)": integer,
"width (p. 591)": integer,
"startTime (p. 591)": "string",
"opacity (p. 592)": integer,
"layer (p. 592)": integer,
"height (p. 592)": integer,
"imageInserterInput (p. 592)": "string"
}
]
}
"antiAlias (p. 636)": enum,
"position (p. 636)": {
  "width (p. 628)": integer,
  "x (p. 628)": integer,
  "y (p. 628)": integer,
  "height (p. 628)": integer
},
"sharpness (p. 636)": integer,
"crop (p. 637)": {
  "width (p. 628)": integer,
  "x (p. 628)": integer,
  "y (p. 628)": integer,
  "height (p. 628)": integer
},
"dropFrameTimecode (p. 637)": enum,
"height (p. 637)": integer
},
"audioDescriptions (p. 622)": [
  {
    "audioTypeControl (p. 526)": enum,
    "languageCodeControl (p. 526)": enum,
    "remixSettings (p. 526)": {
      "channelsOut (p. 628)": integer,
      "channelMapping (p. 628)": {
        "outputChannels (p. 536)": [
          {
            "inputChannels (p. 622)": [integer
          ]
        ]
      },
      "channelsIn (p. 629)": integer
    },
    "audioType (p. 526)": integer,
    "audioSourceName (p. 526)": "string",
    "codecSettings (p. 527)": {
      "codec (p. 525)": enum,
      "wavSettings (p. 525)": {
        "channels (p. 639)": integer,
        "bitDepth (p. 639)": integer,
        "format (p. 640)": enum,
        "sampleRate (p. 640)": integer
      },
      "aacSettings (p. 525)": {
        "vbrQuality (p. 520)": enum,
        "codecProfile (p. 520)": enum,
        "codingMode (p. 520)": enum,
        "specification (p. 521)": enum,
        "bitrate (p. 521)": integer,
        "rawFormat (p. 521)": enum,
        "rateControlMode (p. 521)": enum,
        "sampleRate (p. 521)": integer,
        "audioDescriptionBroadcasterMix (p. 521)": enum
      },
    
  }
]
"ac3Settings (p. 525)" : {
  "dynamicRangeCompressionProfile (p. 522)" : enum,
  "dialnorm (p. 523)" : integer,
  "codingMode (p. 523)" : enum,
  "metadataControl (p. 523)" : enum,
  "bitrate (p. 523)" : integer,
  "lfeFilter (p. 523)" : enum,
  "bitstreamMode (p. 523)" : enum,
  "sampleRate (p. 523)" : integer
},
"aiffSettings (p. 525)" : {
  "channels (p. 524)" : integer,
  "bitDepth (p. 524)" : integer,
  "sampleRate (p. 524)" : integer
},
"eac3Settings (p. 525)" : {
  "dialnorm (p. 553)" : integer,
  "passthroughControl (p. 553)" : enum,
  "metadataControl (p. 553)" : enum,
  "bitrate (p. 553)" : integer,
  "dynamicRangeCompressionRf (p. 553)" : enum,
  "sampleRate (p. 553)" : integer,
  "surroundExMode (p. 554)" : enum,
  "ltRtSurroundMixLevel (p. 554)" : number,
  "dynamicRangeCompressionLine (p. 554)" : enum,
  "lifeControl (p. 554)" : enum,
  "codingMode (p. 554)" : enum,
  "surroundMode (p. 554)" : enum,
  "attenuationControl (p. 554)" : enum,
  "lfeFilter (p. 554)" : enum,
  "phaseControl (p. 554)" : enum,
  "ltRtCenterMixLevel (p. 555)" : number,
  "dcFilter (p. 555)" : enum,
  "bitstreamMode (p. 555)" : enum,
  "stereoDownmix (p. 555)" : enum,
  "loRoSurroundMixLevel (p. 555)" : number,
  "loRoCenterMixLevel (p. 555)" : number
},
"mp2Settings (p. 525)" : {
  "channels (p. 610)" : integer,
  "bitrate (p. 610)" : integer,
  "sampleRate (p. 610)" : integer
},
"languageCode (p. 527)" : enum,
"streamName (p. 527)" : "string",
"audioNormalizationSettings (p. 527)" : {
  "targetLkfs (p. 528)" : number,
  "algorithmControl (p. 528)" : enum,
  "loudnessLogging (p. 528)" : enum,
  "peakCalculation (p. 528)" : enum,
  "correctionGateLevel (p. 529)" : integer,
  "algorithm (p. 529)" : enum
},
"customLanguageCode (p. 527)" : "string"
},
"containerSettings (p. 622)" : {
  "container (p. 542)" : enum,
  "mp4Settings (p. 542)" : {
    "mp4MajorBrand (p. 611)" : "string",
    "moovPlacement (p. 611)" : enum,
    "csigAtom (p. 611)" : enum,
    "freeSpaceBox (p. 611)" : enum
  },
  "m3u8Settings (p. 542)" : {
  
  
  

513


"pmtPid (p. 606)": integer,
"nielsenId3 (p. 606)": enum,
"pcrPid (p. 607)": integer,
"audioPids (p. 607)": [
  integer
],
"audioFramesPerPes (p. 607)": integer,
"scte35Pid (p. 607)": integer,
"transportStreamId (p. 607)": integer,
"pcrControl (p. 607)": enum,
"videoPid (p. 607)": integer,
"privateMetadataPid (p. 608)": integer,
"pmtInterval (p. 608)": integer,
"patInterval (p. 608)": integer,
"programNumber (p. 608)": integer,
"timedMetadataPid (p. 608)": integer,
"timedMetadata (p. 608)": enum,
"scte35Source (p. 609)": enum
},
"m2tsSettings (p. 542)": {
  "dvbTeletextPid (p. 601)": integer,
  "nielsenId3 (p. 601)": enum,
  "bitrate (p. 601)": integer,
  "segmentationTime (p. 601)": number,
  "audioPids (p. 602)": [
    integer
  ],
  "rateMode (p. 602)": enum,
  "maxPcrInterval (p. 602)": integer,
  "audioFramesPerPes (p. 602)": integer,
  "ebpAudioInterval (p. 602)": enum,
  "fragmentTime (p. 602)": number,
  "scte35Pid (p. 602)": integer,
  "pmtInterval (p. 603)": integer,
  "privateMetadataPid (p. 603)": integer,
  "segmentationStyle (p. 603)": enum,
  "audioBufferModel (p. 603)": enum,
  "programNumber (p. 603)": integer,
  "dvbNitSettings (p. 603)": {
    "networkName (p. 546)": "string",
    "networkId (p. 546)": integer,
    "nitInterval (p. 546)": integer
  },
  "timedMetadataPid (p. 603)": integer,
  "scte35Source (p. 604)": enum,
  "pmtPid (p. 604)": integer,
  "bufferModel (p. 604)": integer,
  "ebpPlacement (p. 604)": enum,
  "dvbSdtSettings (p. 604)": {
    "sdtInterval (p. 546)": integer,
    "serviceName (p. 546)": "string",
    "serviceProviderName (p. 547)": "string",
    "outputSdt (p. 547)": enum
  },
  "nullPacketBitrate (p. 604)": number,
  "pcrPid (p. 604)": integer,
  "minEbpInterval (p. 605)": integer,
  "transportStreamId (p. 605)": integer,
  "videoPid (p. 605)": integer,
  "pcrControl (p. 605)": enum,
  "esRateInPes (p. 605)": enum,
  "segmentationMarkers (p. 605)": enum,
  "dvbTdtSettings (p. 605)": {
    "tdtInterval (p. 551)": integer
  },
  "patInterval (p. 605)": integer,
"dvbSubPids (p. 606)" : [
    integer
  ],
"movSettings (p. 542)" : {
  "reference (p. 609)" : enum,
  "paddingControl (p. 610)" : enum,
  "mpeg2FourCCControl (p. 610)" : enum,
  "cs1gAtom (p. 610)" : enum,
  "clapAtom (p. 610)" : enum
},
"f4vSettings (p. 542)" : {
  "moovPlacement (p. 557)" : enum
},
"preset (p. 622)" : "string",
"outputSettings (p. 622)" : {
  "hlsSettings (p. 625)" : {
    "iFrameOnlyManifest (p. 585)" : enum,
    "audioRenditionSets (p. 585)" : "string",
    "audioTrackType (p. 586)" : enum,
    "segmentModifier (p. 586)" : "string",
    "audioGroupId (p. 586)" : "string"
  }
},
"captionDescriptions (p. 622)" : [ {
  "captionSelectorName (p. 533)" : "string",
  "languageDescription (p. 534)" : "string",
  "destinationSettings (p. 534)" : {
    "burninDestinationSettings (p. 534)" : {
      "xPosition (p. 529)" : integer,
      "backgroundColor (p. 530)" : enum,
      "teletextSpacing (p. 530)" : enum,
      "yPosition (p. 530)" : integer,
      "backgroundOpacity (p. 530)" : integer,
      "fontOpacity (p. 530)" : integer,
      "fontResolution (p. 530)" : integer,
      "shadowOpacity (p. 531)" : integer,
      "shadowYOffset (p. 531)" : integer,
      "outlineSize (p. 531)" : integer,
      "outlineColor (p. 531)" : enum,
      "fontSize (p. 531)" : integer,
      "shadowXOffset (p. 531)" : integer,
      "alignment (p. 532)" : enum,
      "shadowColor (p. 532)" : enum,
      "fontColor (p. 532)" : enum
    },
    "teletextDestinationSettings (p. 534)" : {
      "pageNumber (p. 631)" : "string"
    },
    "htmlDestinationSettings (p. 534)" : {
      "stylePassthrough (p. 634)" : enum
    },
    "destinationType (p. 534)" : enum,
    "dvbSubDestinationSettings (p. 535)" : {
      "xPosition (p. 547)" : integer,
      "backgroundColor (p. 547)" : enum,
      "teletextSpacing (p. 547)" : enum,
      "yPosition (p. 547)" : integer,
      "backgroundOpacity (p. 548)" : integer,
      "fontOpacity (p. 548)" : integer,
      "fontResolution (p. 548)" : integer,
      "shadowOpacity (p. 548)" : integer,
      "shadowYOffset (p. 548)" : integer,
      "outlineSize (p. 548)" : integer,
"outlineColor (p. 549)": enum,
"fontSize (p. 549)": integer,
"shadowXOffset (p. 549)": integer,
"alignment (p. 549)": enum,
"shadowColor (p. 549)": enum,
"fontColor (p. 549)": enum
},
"sccDestinationSettings (p. 535)": {
"framerate (p. 629)": enum
},
"languageCode (p. 534)": enum,
"customLanguageCode (p. 534)": "string"
],
"nameModifier (p. 622)": "string"
},
"outputGroupSettings (p. 623)": {
"dashIsoGroupSettings (p. 624)": {
"fragmentLength (p. 543)": integer,
"baseUrl (p. 543)": "string",
"minBufferTime (p. 543)": integer,
"encryption (p. 544)": {
"spekeKeyProvider (p. 543)": {
"resourceId (p. 630)": "string",
"systemIds (p. 630)": [
"string"
],
"url (p. 630)": "string"
}
},
"destination (p. 544)": "string",
"segmentLength (p. 544)": integer,
"segmentControl (p. 544)": enum,
"hbbtvCompliance (p. 544)": enum
},
"fileGroupSettings (p. 624)": {
"destination (p. 557)": "string"
},
"msSmoothGroupSettings (p. 624)": {
"fragmentLength (p. 619)": integer,
"encryption (p. 619)": {
"spekeKeyProvider (p. 618)": {
"resourceId (p. 630)": "string",
"systemIds (p. 630)": [
"string"
],
"url (p. 630)": "string"
}
},
"audioDeduplication (p. 619)": enum,
"manifestEncoding (p. 619)": enum,
"destination (p. 619)": "string"
},
"cmafGroupSettings (p. 624)": {
"writeHlsManifest (p. 537)": enum,
"writeDashManifest (p. 537)": enum,
"fragmentLength (p. 537)": integer,
"streamInfResolution (p. 538)": enum,
"minBufferTime (p. 538)": integer,
"clientCache (p. 538)": enum,
"codecSpecification (p. 538)": enum,
"destination (p. 538)": "string",
"segmentControl (p. 538)": enum,
"baseUrl (p. 538)": "string",
encryption (p. 538): {
  initializationVectorInManifest (p. 536): enum,
  constantInitializationVector (p. 537): "string",
  staticKeyProvider (p. 537): {
    keyFormatVersions (p. 630): "string",
    keyFormat (p. 630): "string",
    staticKeyValue (p. 630): "string",
    url (p. 631): "string"
  },
  type (p. 537): enum,
  encryptionMethod (p. 537): enum
},
manifestCompression (p. 539): enum,
segmentLength (p. 539): integer,
manifestDurationFormat (p. 539): enum
},
type (p. 624): enum,
hlsGroupSettings (p. 624): {
  segmentsPerSubdirectory (p. 581): integer,
  streamInfResolution (p. 581): enum,
  timestampDeltaMilliseconds (p. 582): integer,
  outputSelection (p. 582): enum,
  captionLanguageMappings (p. 582): [
    languageDescription (p. 579): "string",
    captionChannel (p. 579): integer,
    languageCode (p. 579): enum,
    customLanguageCode (p. 580): "string"
  ],
  clientCache (p. 582): enum,
  codecSpecification (p. 582): enum,
  destination (p. 582): "string",
  timedMetadataId3Frame (p. 582): enum,
  segmentControl (p. 582): enum,
  timedMetadataId3Period (p. 583): integer,
  captionLanguageSetting (p. 583): enum,
  minSegmentLength (p. 583): integer,
  directoryStructure (p. 583): enum,
  programDateTime (p. 583): enum,
  baseUrl (p. 583): "string",
  encryption (p. 583): {
    initializationVectorInManifest (p. 580): enum,
    constantInitializationVector (p. 581): "string",
    staticKeyProvider (p. 581): {
      keyFormatVersions (p. 630): "string",
      keyFormat (p. 630): "string",
      staticKeyValue (p. 630): "string",
      url (p. 631): "string"
    },
    type (p. 581): enum,
    encryptionMethod (p. 581): enum,
    spekeKeyProvider (p. 581): {
      resourceId (p. 630): "string",
      systemIds (p. 630): [
        "string"
      ],
      url (p. 630): "string"
    }
  },
adMarkers (p. 583): [
  enum
],
programDateTimePeriod (p. 584): integer,
manifestCompression (p. 584): enum,
segmentLength (p. 584): integer,
"manifestDurationFormat (p. 584)" : enum,
),
"name (p. 623)" : "string",
"customName (p. 623)" : "string"
}
],
"nielsenConfiguration (p. 595)" : {
  "distributorId (p. 619)" : "string",
  "breakoutCode (p. 620)" : integer
},
"inputs (p. 595)" : [
  {
    "audioSelectors (p. 587)" : {
    
    },
    "audioSelectorGroups (p. 587)" : {
    
    },
    "videoSelector (p. 587)" : {
      "colorSpace (p. 638)" : enum,
      "hdr10Metadata (p. 638)" : {
        "redPrimaryY (p. 576)" : integer,
        "greenPrimaryY (p. 577)" : integer,
        "whitePointX (p. 577)" : integer,
        "maxLuminance (p. 577)" : integer,
        "greenPrimaryX (p. 577)" : integer,
        "whitePointY (p. 577)" : integer,
        "redPrimaryX (p. 578)" : integer,
        "bluePrimaryX (p. 578)" : integer,
        "bluePrimaryY (p. 578)" : integer,
        "maxFrameAverageLightLevel (p. 578)" : integer,
        "maxContentLightLevel (p. 578)" : integer,
        "minLuminance (p. 579)" : integer
      },
      "programNumber (p. 638)" : integer,
      "pid (p. 639)" : integer,
      "colorSpaceUsage (p. 639)" : enum
    },
    "denoiseFilter (p. 587)" : enum,
    "filterEnable (p. 587)" : enum,
    "deblockFilter (p. 587)" : enum,
    "filterStrength (p. 588)" : integer,
    "programNumber (p. 588)" : integer,
    "timecodeSource (p. 588)" : enum,
    "captionSelectors (p. 588)" : {
    
    },
    "fileInput (p. 588)" : "string",
    "inputClippings (p. 588)" : [ {
      "startTimecode (p. 589)" : "string",
      "endTimecode (p. 589)" : "string"
    } ],
    "psiControl (p. 589)" : enum
  },
],
"timedMetadataInsertion (p. 595)" : {
  "id3Insertions (p. 633)" : [ {
    "id3 (p. 586)" : "string",
    "timecode (p. 586)" : "string"
  } ],
  "availBlanking (p. 595)" : { {
    "availBlankingImage (p. 529)" : "string" 
  } }
},
  "role (p. 592)": "string",
  "outputGroupDetails (p. 593)": [
    {"outputDetails (p. 623)": [
      {"durationInMs (p. 623)": integer,
       "videoDetails (p. 623)": {
         "heightInPx (p. 637)": integer,
         "widthInPx (p. 637)": integer
       }
    ]
  }]
},
  "jobTemplate (p. 593)": "string",
  "timing (p. 593)": {
    "finishTime (p. 633)": "string",
    "submitTime (p. 634)": "string",
    "startTime (p. 634)": "string"
  },
  "errorMessage (p. 593)": "string",
  "errorCode (p. 593)": integer,
  "createdAt (p. 593)": "string",
  "id (p. 593)": "string",
  "arn (p. 593)": "string",
  "userMetadata (p. 594)": {
    "queue (p. 594)": "string",
    "status (p. 594)": enum
  }
}

Example CancelJobResponse

{}

Example ExceptionBody

{
  "message (p. 557)": "string"
}

Properties

AacAudioDescriptionBroadcasterMix (enum)

Choose BROADCASTER_MIXED_AD when the input contains pre-mixed main audio + audio description (AD) as a stereo pair. The value for AudioType will be set to 3, which signals to downstream systems that this stream contains “broadcaster mixed AD”. Note that the input received by the encoder must contain pre-mixed audio; the encoder does not perform the mixing. When you choose BROADCASTER_MIXED_AD, the encoder ignores any values you provide in AudioType and FollowInputAudioType. Choose NORMAL when the input does not contain pre-mixed audio + audio description (AD). In this case, the encoder will use any values you provide for AudioType and FollowInputAudioType.

BROADCASTER_MIXED_AD
NORMAL
AacCodecProfile (enum)

AAC Profile.

- LC
- HEV1
- HEV2

AacCodingMode (enum)

Mono (Audio Description), Mono, Stereo, or 5.1 channel layout. Valid values depend on rate control mode and profile. "1.0 - Audio Description (Receiver Mix)" setting receives a stereo description plus control track and emits a mono AAC encode of the description track, with control data emitted in the PES header as per ETSI TS 101 154 Annex E.

- AD_RECEIVER_MIX
- CODING_MODE_1_0
- CODING_MODE_1_1
- CODING_MODE_2_0
- CODING_MODE_5_1

AacRateControlMode (enum)

Rate Control Mode.

- CBR
- VBR

AacRawFormat (enum)

 Enables LATM/LOAS AAC output. Note that if you use LATM/LOAS AAC in an output, you must choose "No container" for the output container.

- LATM_LOAS
- NONE

AacSettings

vbrQuality

- Type: string
- Required: False

codecProfile

- Type: string
- Required: False

codingMode

- Type: string
- Required: True
specification

  Type: string
  Required: False

bitrate

Average bitrate in bits/second. Defaults and valid values depend on rate control mode and profile.

  Type: integer
  Required: False
  Minimum: 6000
  Maximum: 1024000

rawFormat

  Type: string
  Required: False

rateControlMode

  Type: string
  Required: False

sampleRate

Sample rate in Hz. Valid values depend on rate control mode and profile.

  Type: integer
  Required: True
  Minimum: 8000
  Maximum: 96000

audioDescriptionBroadcasterMix

  Type: string
  Required: False

AacSpecification (enum)

Use MPEG-2 AAC instead of MPEG-4 AAC audio for raw or MPEG-2 Transport Stream containers.

  MPEG2
  MPEG4

AacVbrQuality (enum)

VBR Quality Level - Only used if rate_control_mode is VBR.

  LOW
  MEDIUM_LOW
  MEDIUM_HIGH
  HIGH
**Ac3BitstreamMode (enum)**

Specifies the "Bitstream Mode" (bsmod) for the emitted AC-3 stream. See ATSC A/52-2012 for background on these values.

- COMPLETE_MAIN
- COMMENTARY
- DIALOGUE
- EMERGENCY
- HEARING_IMPAIRED
- MUSIC_AND_EFFECTS
- VISUALLY_IMPAIRED
- VOICE_OVER

**Ac3CodingMode (enum)**

Dolby Digital coding mode. Determines number of channels.

- CODING_MODE_1_0
- CODING_MODE_1_1
- CODING_MODE_2_0
- CODING_MODE_3_2_LFE

**Ac3DynamicRangeCompressionProfile (enum)**

If set to FILM_STANDARD, adds dynamic range compression signaling to the output bitstream as defined in the Dolby Digital specification.

- FILM_STANDARD
- NONE

**Ac3LfeFilter (enum)**

Applies a 120Hz lowpass filter to the LFE channel prior to encoding. Only valid with 3_2_LFE coding mode.

- ENABLED
- DISABLED

**Ac3MetadataControl (enum)**

When set to FOLLOW_INPUT, encoder metadata will be sourced from the DD, DD+, or DolbyE decoder that supplied this audio data. If audio was not supplied from one of these streams, then the static metadata settings will be used.

- FOLLOW_INPUT
- USE_CONFIGURED

**Ac3Settings**

**dynamicRangeCompressionProfile**

- **Type**: string
- **Required**: False
dialnorm

Sets the dialnorm for the output. If blank and input audio is Dolby Digital, dialnorm will be passed through.

Type: integer
Required: False
Minimum: 1
Maximum: 31

codingMode

Type: string
Required: False

metadataControl

Type: string
Required: False

bitrate

Average bitrate in bits/second. Valid bitrates depend on the coding mode.

Type: integer
Required: False
Minimum: 64000
Maximum: 640000

lfeFilter

Type: string
Required: False

bitstreamMode

Type: string
Required: False

sampleRate

Sample rate in hz. Sample rate is always 48000.

Type: integer
Required: False
Minimum: 48000
Maximum: 48000

AfdSignaling (enum)

This setting only applies to H.264 and MPEG2 outputs. Use Insert AFD signaling (AfdSignaling) to specify whether the service includes AFD values in the output video data and what those values are. * Choose None to remove all AFD values from this output. * Choose Fixed to ignore input AFD values and instead
encode the value specified in the job. * Choose Auto to calculate output AFD values based on the input AFD scaler data.

NONE
AUTO
FIXED

AiffSettings

channels
Set Channels to specify the number of channels in this output audio track. Choosing Mono in the console will give you 1 output channel; choosing Stereo will give you 2. In the API, valid values are 1 and 2.

Type: integer
Required: False
Minimum: 1
Maximum: 2

bitDepth
Specify Bit depth (BitDepth), in bits per sample, to choose the encoding quality for this audio track.

Type: integer
Required: False
Minimum: 16
Maximum: 24

sampleRate
Sample rate in hz.

Type: integer
Required: False
Minimum: 8000
Maximum: 192000

AncillarySourceSettings

sourceAncillaryChannelNumber
Specifies the 608 channel number in the ancillary data track from which to extract captions. Unused for passthrough.

Type: integer
Required: False
Minimum: 1
Maximum: 4

AntiAlias (enum)
Enable Anti-alias (AntiAlias) to enhance sharp edges in video output when your input resolution is much larger than your output resolution. Default is enabled.
DISABLED
ENABLED

**AudioCodec (enum)**

Type of Audio codec.

AAC
MP2
WAV
AIFF
AC3
EAC3
PASSTHROUGH

**AudioCodecSettings**

codec

Type: string  
Required: True

wavSettings

Type: WavSettings (p. 639)  
Required: False

aacSettings

Type: AacSettings (p. 520)  
Required: False

ac3Settings

Type: Ac3Settings (p. 522)  
Required: False

aiffSettings

Type: AiffSettings (p. 524)  
Required: False

eac3Settings

Type: Eac3Settings (p. 553)  
Required: False

mp2Settings

Type: Mp2Settings (p. 610)
Properties

**Required**: False

**AudioDefaultSelection (enum)**

Enable this setting on one audio selector to set it as the default for the job. The service uses this default for outputs where it can't find the specified input audio. If you don't set a default, those outputs have no audio.

- DEFAULT
- NOT_DEFAULT

**AudioDescription**

**audioTypeControl**

- **Type**: string
- **Required**: False

**languageCodeControl**

- **Type**: string
- **Required**: False

**remixSettings**

Advanced audio remixing settings.

- **Type**: RemixSettings (p. 628)
- **Required**: False

**audioType**

Applies only if Follow Input Audio Type is unchecked (false). A number between 0 and 255. The following are defined in ISO-IEC 13818-1: 0 = Undefined, 1 = Clean Effects, 2 = Hearing Impaired, 3 = Visually Impaired Commentary, 4-255 = Reserved.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 255

**audioSourceName**

Specifies which audio data to use from each input. In the simplest case, specify an "Audio Selector":#inputs-audio_selector by name based on its order within each input. For example if you specify "Audio Selector 3", then the third audio selector will be used from each input. If an input does not have an "Audio Selector 3", then the audio selector marked as "default" in that input will be used. If there is no audio selector marked as "default", silence will be inserted for the duration of that input. Alternatively, an "Audio Selector Group":#inputs-audio_selector_group name may be specified, with similar default/silence behavior. If no audio_source_name is specified, then "Audio Selector 1" will be chosen automatically.

- **Type**: string
Properties

Required: False

codecSettings

Type: AudioCodecSettings (p. 525)
Required: True

languageCode

Indicates the language of the audio output track. The ISO 639 language specified in the 'Language Code' drop down will be used when 'Follow Input Language Code' is not selected or when 'Follow Input Language Code' is selected but there is no ISO 639 language code specified by the input.

Type: string
Required: False

streamName

Used for MS Smooth and Apple HLS outputs. Indicates the name displayed by the player (eg. English, or Director Commentary). Alphanumeric characters, spaces, and underscore are legal.

Type: string
Required: False
Pattern: ^[\w\s]*$

audioNormalizationSettings

Type: AudioNormalizationSettings (p. 528)
Required: False

customLanguageCode

Specify the language for this audio output track, using the ISO 639-2 or ISO 639-3 three-letter language code. The language specified will be used when 'Follow Input Language Code' is not selected or when 'Follow Input Language Code' is selected but there is no ISO 639 language code specified by the input.

Type: string
Required: False
Pattern: ^[A-Za-z]{3}$

AudioLanguageCodeControl (enum)

Choosing FOLLOW_INPUT will cause the ISO 639 language code of the output to follow the ISO 639 language code of the input. The language specified for languageCode' will be used when USE_CONFIGURED is selected or when FOLLOW_INPUT is selected but there is no ISO 639 language code specified by the input.

FOLLOW_INPUT
USE_CONFIGURED

AudioNormalizationAlgorithm (enum)

Audio normalization algorithm to use. 1770-1 conforms to the CALM Act specification, 1770-2 conforms to the EBU R-128 specification.
AudioNormalizationAlgorithmControl (enum)
When enabled the output audio is corrected using the chosen algorithm. If disabled, the audio will be measured but not adjusted.

    CORRECT_AUDIO
    MEASURE_ONLY

AudioNormalizationLoudnessLogging (enum)
If set to LOG, log each output's audio track loudness to a CSV file.

    LOG
    DONT_LOG

AudioNormalizationPeakCalculation (enum)
If set to TRUE_PEAK, calculate and log the TruePeak for each output's audio track loudness.

    TRUE_PEAK
    NONE

AudioNormalizationSettings

targetLkfs
Target LKFS(loudness) to adjust volume to. If no value is entered, a default value will be used according to the chosen algorithm. The CALM Act (1770-1) recommends a target of -24 LKFS. The EBU R-128 specification (1770-2) recommends a target of -23 LKFS.

    Type: number
    Required: False
    Format: float
    Minimum: -59.0
    Maximum: 0.0

algorithmControl

    Type: string
    Required: False

loudnessLogging

    Type: string
    Required: False

peakCalculation

    Type: string
    Required: False
**correctionGateLevel**

Content measuring above this level will be corrected to the target level. Content measuring below this level will not be corrected. Gating only applies when not using `real_time_correction`.

- **Type**: integer
- **Required**: False
- **Minimum**: -70
- **Maximum**: 0

**algorithm**

- **Type**: string
- **Required**: False

**AudioSelectorType (enum)**

Specifies the type of the audio selector.

- **PID**
- **TRACK**
- **LANGUAGE_CODE**

**AudioTypeControl (enum)**

When set to `FOLLOW_INPUT`, if the input contains an ISO 639 audio_type, then that value is passed through to the output. If the input contains no ISO 639 audio_type, the value in Audio Type is included in the output. Otherwise the value in Audio Type is included in the output. Note that this field and `audioType` are both ignored if `audioDescriptionBroadcasterMix` is set to `BROADCASTER_MIXED_AD`.

- **FOLLOW_INPUT**
- **USE_CONFIGURED**

**AvailBlanking**

**availBlankingImage**

Blanking image to be used. Leave empty for solid black. Only bmp and png images are supported.

- **Type**: string
- **Required**: False
- **Pattern**: `^(s3://)(.*?)(bmp|BMP|png|PNG)$`

**BurninDestinationSettings**

**xPosition**

Specifies the horizontal position of the caption relative to the left side of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the left of the output. If no explicit x_position is provided, the horizontal caption position will be determined by the alignment parameter. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
**Properties**

- **backgroundColor**
  - Type: string
  - Required: False

- **teletextSpacing**
  - Type: string
  - Required: False

- **yPosition**
  Specifies the vertical position of the caption relative to the top of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the top of the output. If no explicit y_position is provided, the caption will be positioned towards the bottom of the output. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.
  - Type: integer
  - Required: False
  - Minimum: 0
  - Maximum: 2147483647

- **backgroundOpacity**
  Specifies the opacity of the background rectangle. 255 is opaque; 0 is transparent. Leaving this parameter blank is equivalent to setting it to 0 (transparent). All burn-in and DVB-Sub font settings must match.
  - Type: integer
  - Required: False
  - Minimum: 0
  - Maximum: 255

- **fontOpacity**
  Specifies the opacity of the burned-in captions. 255 is opaque; 0 is transparent. All burn-in and DVB-Sub font settings must match.
  - Type: integer
  - Required: True
  - Minimum: 0
  - Maximum: 255

- **fontResolution**
  Font resolution in DPI (dots per inch); default is 96 dpi. All burn-in and DVB-Sub font settings must match.
  - Type: integer
**Properties**

**shadowOpacity**

Specifies the opacity of the shadow. 255 is opaque; 0 is transparent. Leaving this parameter blank is equivalent to setting it to 0 (transparent). All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 255

**shadowYOffset**

Specifies the vertical offset of the shadow relative to the captions in pixels. A value of -2 would result in a shadow offset 2 pixels above the text. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: False
- **Minimum**: -2147483648
- **Maximum**: 2147483647

**outlineSize**

Specifies font outline size in pixels. This option is not valid for source captions that are either 608/ embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: True
- **Minimum**: 0
- **Maximum**: 10

**outlineColor**

- **Type**: string
- **Required**: True

**fontSize**

A positive integer indicates the exact font size in points. Set to 0 for automatic font size selection. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 96

**shadowXOffset**

Specifies the horizontal offset of the shadow relative to the captions in pixels. A value of -2 would result in a shadow offset 2 pixels to the left. All burn-in and DVB-Sub font settings must match.
**Type**: integer  
**Required**: False  
**Minimum**: -2147483648  
**Maximum**: 2147483647

**alignment**

**Type**: string  
**Required**: True

**shadowColor**

**Type**: string  
**Required**: False

**fontColor**

**Type**: string  
**Required**: False

**BurninSubtitleAlignment (enum)**

If no explicit x_position or y_position is provided, setting alignment to centered will place the captions at the bottom center of the output. Similarly, setting a left alignment will align captions to the bottom left of the output. If x and y positions are given in conjunction with the alignment parameter, the font will be justified (either left or centered) relative to those coordinates. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- CENTERED
- LEFT

**BurninSubtitleBackgroundColor (enum)**

Specifies the color of the rectangle behind the captions. All burn-in and DVB-Sub font settings must match.

- NONE
- BLACK
- WHITE

**BurninSubtitleFontColor (enum)**

Specifies the color of the burned-in captions. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- WHITE
- BLACK
- YELLOW
- RED
- GREEN
- BLUE
BurninSubtitleOutlineColor (enum)

Specifies font outline color. This option is not valid for source captions that are either 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- BLACK
- WHITE
- YELLOW
- RED
- GREEN
- BLUE

BurninSubtitleShadowColor (enum)

Specifies the color of the shadow cast by the captions. All burn-in and DVB-Sub font settings must match.

- NONE
- BLACK
- WHITE

BurninSubtitleTeletextSpacing (enum)

Only applies to jobs with input captions in Teletext or STL formats. Specify whether the spacing between letters in your captions is set by the captions grid or varies depending on letter width. Choose fixed grid to conform to the spacing specified in the captions file more accurately. Choose proportional to make the text easier to read if the captions are closed caption.

- FIXED_GRID
- PROPORTIONAL

CancelJobRequest

id

The Job ID of the job to be cancelled.

- Type: string
- Required: False

CancelJobResponse

CaptionDescription

captionSelectorName

Specifies which "Caption Selector":#inputs-captions_selector to use from each input when generating captions. The name should be of the format "Caption Selector <N>", which denotes that the Nth Caption Selector will be used from each input.

- Type: string
Required: True

**languageDescription**

Human readable information to indicate captions available for players (eg. English, or Spanish). Alphanumeric characters, spaces, and underscore are legal.

- **Type:** string
- **Required:** False
- **Pattern:** `^[\w ]*$`

**destinationSettings**

- **Type:** CaptionDestinationSettings (p. 534)
- **Required:** True

**languageCode**

Indicates the language of the caption output track.

- **Type:** string
- **Required:** False

**customLanguageCode**

Indicates the language of the caption output track, using the ISO 639-2 or ISO 639-3 three-letter language code.

- **Type:** string
- **Required:** False
- **Pattern:** `^[A-Za-z]{3}$`

**CaptionDestinationSettings**

**burninDestinationSettings**

- **Type:** BurninDestinationSettings (p. 529)
- **Required:** False

**teletextDestinationSettings**

- **Type:** TeletextDestinationSettings (p. 631)
- **Required:** False

**ttmlDestinationSettings**

- **Type:** TtmlDestinationSettings (p. 634)
- **Required:** False

**destinationType**

- **Type:** string
Required: True

dvbSubDestinationSettings
  Type: DvbSubDestinationSettings (p. 547)
  Required: False

cccDestinationSettings
  Type: SccDestinationSettings (p. 629)
  Required: False

CaptionDestinationType (enum)
Type of Caption output, including Burn-In, Embedded, SCC, SRT, TTML, WebVTT, DVB-Sub, Teletext.
  BURN_IN
  DVB_SUB
  EMBEDDED
  SCC
  SRT
  TELETEXT
  TTML
  WEBVTT

CaptionSourceSettings

fileSourceSettings
  Type: FileSourceSettings (p. 557)
  Required: False

ancillarySourceSettings
  Type: AncillarySourceSettings (p. 524)
  Required: False

embeddedSourceSettings
  Type: EmbeddedSourceSettings (p. 556)
  Required: False

sourceType
  Type: string
  Required: True

dvbSubSourceSettings
  Type: DvbSubSourceSettings (p. 549)
Required: False

teletextSourceSettings

Type: TeletextSourceSettings (p. 631)
Required: False

CaptionSourceType (enum)

Use Source (SourceType) to identify the format of your input captions. The service cannot auto-detect caption format.

ANCILLARY
DVB_SUB
EMBEDDED
SCC
TTML
STL
SRT
TELETEXT
NULL_SOURCE

ChannelMapping

outputChannels

List of output channels

Type: Array of type OutputChannelMapping (p. 622)
Required: True

CmafClientCache (enum)

When set to ENABLED, sets #EXT-X-ALLOW-CACHE:no tag, which prevents client from saving media segments for later replay.

DISABLED
ENABLED

CmafCodecSpecification (enum)

Specification to use (RFC-6381 or the default RFC-4281) during m3u8 playlist generation.

RFC_6381
RFC_4281

CmafEncryptionSettings

initializationVectorInManifest

Type: string
Required: False
constantInitializationVector
This is a 128-bit, 16-byte hex value represented by a 32-character text string. If this parameter is not set then the Initialization Vector will follow the segment number by default.

  Type: string
  Required: False
  Pattern: ^[0-9a-fA-F]{32}$

staticKeyProvider

  Type: StaticKeyProvider (p. 630)
  Required: False

type

  Type: string
  Required: True

encryptionMethod

  Type: string
  Required: False

CmafEncryptionType (enum)
Encrypts the segments with the given encryption scheme. Leave blank to disable. Selecting 'Disabled' in the web interface also disables encryption.

  SAMPLE_AES

CmafGroupSettings

writeHlsManifest

  Type: string
  Required: False

writeDashManifest

  Type: string
  Required: False

fragmentLength
Length of fragments to generate (in seconds). Fragment length must be compatible with GOP size and Framerate. Note that fragments will end on the next keyframe after this number of seconds, so actual fragment length may be longer. When Emit Single File is checked, the fragmentation is internal to a single output file and it does not cause the creation of many output files as in other output types.

  Type: integer
  Required: True
  Minimum: 1
Maximum: 2147483647

**streamInfResolution**

Type: string  
Required: False  

**minBufferTime**

Minimum time of initially buffered media that is needed to ensure smooth playout.

Type: integer  
Required: False  
Minimum: 0  
Maximum: 2147483647

**clientCache**

Type: string  
Required: False

**codecSpecification**

Type: string  
Required: False

**destination**

Use Destination (Destination) to specify the S3 output location and the output filename base. Destination accepts format identifiers. If you do not specify the base filename in the URI, the service will use the filename of the input file. If your job has multiple inputs, the service uses the filename of the first input file.

Type: string  
Required: False  
Pattern: ^s3:/\/*\/

**segmentControl**

Type: string  
Required: False

**baseUrl**

A partial URI prefix that will be put in the manifest file at the top level BaseURL element. Can be used if streams are delivered from a different URL than the manifest file.

Type: string  
Required: False

**encryption**

DRM settings.
Type: CmafEncryptionSettings (p. 536)  
Required: False

**manifestCompression**

- **Type:** string  
- **Required:** False

**segmentLength**

Use this setting to specify the length, in seconds, of each individual CMAF segment. This value applies to the whole package; that is, to every output in the output group. Note that segments end on the first keyframe after this number of seconds, so the actual segment length might be slightly longer. If you set Segment control (CmafSegmentControl) to single file, the service puts the content of each output in a single file that has metadata that marks these segments. If you set it to segmented files, the service creates multiple files for each output, each with the content of one segment.

- **Type:** integer  
- **Required:** True  
- **Minimum:** 1  
- **Maximum:** 2147483647

**manifestDurationFormat**

- **Type:** string  
- **Required:** False

**CmafInitializationVectorInManifest (enum)**

The Initialization Vector is a 128-bit number used in conjunction with the key for encrypting blocks. If set to INCLUDE, Initialization Vector is listed in the manifest. Otherwise Initialization Vector is not in the manifest.

- INCLUDE  
- EXCLUDE

**CmafKeyProviderType (enum)**

Indicates which type of key provider is used for encryption.

- STATIC_KEY

**CmafManifestCompression (enum)**

When set to GZIP, compresses HLS playlist.

- GZIP  
- NONE

**CmafManifestDurationFormat (enum)**

Indicates whether the output manifest should use floating point values for segment duration.
FLOATING_POINT
INTEGER

CmafSegmentControl (enum)

When set to SINGLE_FILE, a single output file is generated, which is internally segmented using the Fragment Length and Segment Length. When set to SEGMENTED_FILES, separate segment files will be created.

SINGLE_FILE
SEGMENTED_FILES

CmafStreamInfResolution (enum)

Include or exclude RESOLUTION attribute for video in EXT-X-STREAM-INF tag of variant manifest.

INCLUDE
EXCLUDE

CmafWriteDASHManifest (enum)

When set to ENABLED, a DASH MPD manifest will be generated for this output.

DISABLED
ENABLED

CmafWriteHLSManifest (enum)

When set to ENABLED, an Apple HLS manifest will be generated for this output.

DISABLED
ENABLED

ColorCorrector

saturation

Saturation level.

Type: integer
Required: False
Minimum: 1
Maximum: 100

brightness

Brightness level.

Type: integer
Required: False
Minimum: 1
Maximum: 100
**hdr10Metadata**

Type: Hdr10Metadata (p. 576)
Required: False

**contrast**

Contrast level.
Type: integer
Required: False
Minimum: 1
Maximum: 100

**hue**

Hue in degrees.
Type: integer
Required: False
Minimum: -180
Maximum: 180

**colorSpaceConversion**

Type: string
Required: False

**ColorMetadata (enum)**

Enable insert color metadata (ColorMetadata) to include color metadata in this output. This setting is enabled by default.

- IGNORE
- INSERT

**ColorSpace (enum)**

If your input video has accurate color space metadata, or if you don’t know about color space, leave this set to the default value FOLLOW. The service will automatically detect your input color space. If your input video has metadata indicating the wrong color space, or if your input video is missing color space metadata that should be there, specify the accurate color space here. If you choose HDR10, you can also correct inaccurate color space coefficients, using the HDR master display information controls. You must also set Color space usage (ColorSpaceUsage) to FORCE for the service to use these values.

- FOLLOW
- REC_601
- REC_709
- HDR10
- HLG_2020

**ColorSpaceConversion (enum)**

Determines if colorspace conversion will be performed. If set to _None_, no conversion will be performed. If _Force 601_ or _Force 709_ are selected, conversion will be performed for inputs with
differing colorspaces. An input's colorspace can be specified explicitly in the "Video Selector": #inputs-video_selector if necessary.

NONE
FORCE_601
FORCE_709
FORCE_HDR10
FORCE_HLG_2020

ColorSpaceUsage (enum)

There are two sources for color metadata, the input file and the job configuration (in the Color space and HDR master display information settings). The Color space usage setting controls which takes precedence. FORCE: The system will use color metadata supplied by user, if any. If the user does not supply color metadata, the system will use data from the source. FALLBACK: The system will use color metadata from the source. If source has no color metadata, the system will use user-supplied color metadata values if available.

FORCE
FALLBACK

ContainerSettings

container

Type: string
Required: True

mp4Settings

Type: Mp4Settings (p. 611)
Required: False

m3u8Settings

Type: M3u8Settings (p. 606)
Required: False

m2tsSettings

Type: M2tsSettings (p. 601)
Required: False

movSettings

Type: MovSettings (p. 609)
Required: False

f4vSettings

Type: F4vSettings (p. 557)
Required: False
**ContainerType (enum)**

Container for this output. Some containers require a container settings object. If not specified, the default object will be created.

- F4V
- ISMV
- M2TS
- M3U8
- CMFC
- MOV
- MP4
- MPD
- MXF
- RAW

**DashIsoEncryptionSettings**

*spekeKeyProvider*

*Type: SpekeKeyProvider (p. 630)*

*Required: True*

**DashIsoGroupSettings**

*fragmentLength*

Length of fragments to generate (in seconds). Fragment length must be compatible with GOP size and Framerate. Note that fragments will end on the next keyframe after this number of seconds, so actual fragment length may be longer. When Emit Single File is checked, the fragmentation is internal to a single output file and it does not cause the creation of many output files as in other output types.

*Type: integer*

*Required: True*

*Minimum: 1*

*Maximum: 2147483647*

*baseUrl*

A partial URI prefix that will be put in the manifest (.mpd) file at the top level BaseURL element. Can be used if streams are delivered from a different URL than the manifest file.

*Type: string*

*Required: False*

**minBufferTime**

Minimum time of initially buffered media that is needed to ensure smooth playout.

*Type: integer*

*Required: False*

*Minimum: 0*
**Maximum**: 2147483647

**encryption**

DRM settings.

*Type*: DashIsoEncryptionSettings (p. 543)

*Required*: False

**destination**

Use Destination (Destination) to specify the S3 output location and the output filename base. Destination accepts format identifiers. If you do not specify the base filename in the URI, the service will use the filename of the input file. If your job has multiple inputs, the service uses the filename of the first input file.

*Type*: string

*Required*: False

*Pattern*: ^s3:/\//

**segmentLength**

Length of mpd segments to create (in seconds). Note that segments will end on the next keyframe after this number of seconds, so actual segment length may be longer. When Emit Single File is checked, the segmentation is internal to a single output file and it does not cause the creation of many output files as in other output types.

*Type*: integer

*Required*: True

*Minimum*: 1

*Maximum*: 2147483647

**segmentControl**

*Type*: string

*Required*: False

**hbbtvCompliance**

*Type*: string

*Required*: False

**DashIsoHbbtvCompliance (enum)**

Supports HbbTV specification as indicated

- HBBTV_1_5
- NONE

**DashIsoSegmentControl (enum)**

When set to SINGLE_FILE, a single output file is generated, which is internally segmented using the Fragment Length and Segment Length. When set to SEGMENTED_FILES, separate segment files will be created.
SINGLE_FILE
SEGMENTED_FILES

**DeinterlaceAlgorithm (enum)**

Only applies when you set Deinterlace (DeinterlaceMode) to Deinterlace (DEINTERLACE) or Adaptive (ADAPTIVE). Motion adaptive interpolate (INTERPOLATE) produces sharper pictures, while blend (BLEND) produces smoother motion. Use (INTERPOLATE_TICKER) OR (BLEND_TICKER) if your source file includes a ticker, such as a scrolling headline at the bottom of the frame.

INTERPOLATE
INTERPOLATE_TICKER
BLEND
BLEND_TICKER

**Deinterlacer**

**mode**

*Type: string*

*Required: False*

**control**

*Type: string*

*Required: False*

**algorithm**

*Type: string*

*Required: False*

**DeinterlacerControl (enum)**

- When set to NORMAL (default), the deinterlacer does not convert frames that are tagged in metadata as progressive. It will only convert those that are tagged as some other type. - When set to FORCE_ALL_FRAMES, the deinterlacer converts every frame to progressive - even those that are already tagged as progressive. Turn Force mode on only if there is a good chance that the metadata has tagged frames as progressive when they are not progressive. Do not turn on otherwise; processing frames that are already progressive into progressive will probably result in lower quality video.

FORCE_ALL_FRAMES
NORMAL

**DeinterlacerMode (enum)**

Use Deinterlacer (DeinterlaceMode) to choose how the service will do deinterlacing. Default is Deinterlace. - Deinterlace converts interlaced to progressive. - Inverse telecine converts Hard Telecine 29.97i to progressive 23.976p. - Adaptive auto-detects and converts to progressive.

DEINTERLACE
INVERSE_TELECINE
ADAPTIVE
DropFrameTimecode (enum)

Applies only to 29.97 fps outputs. When this feature is enabled, the service will use drop-frame timecode on outputs. If it is not possible to use drop-frame timecode, the system will fall back to non-drop-frame. This setting is enabled by default when Timecode insertion (TimecodeInsertion) is enabled.

DISPLAYED
ENABLED

DvbNitSettings

networkName

The network name text placed in the network_name_descriptor inside the Network Information Table. Maximum length is 256 characters.

Type: string
Required: True

networkId

The numeric value placed in the Network Information Table (NIT).

Type: integer
Required: True
Minimum: 0
Maximum: 65535

nitInterval

The number of milliseconds between instances of this table in the output transport stream.

Type: integer
Required: True
Minimum: 25
Maximum: 10000

DvbSdtSettings

sdtInterval

The number of milliseconds between instances of this table in the output transport stream.

Type: integer
Required: False
Minimum: 25
Maximum: 2000

serviceName

The service name placed in the service_descriptor in the Service Description Table. Maximum length is 256 characters.
Properties

**Type**: string
**Required**: False

**serviceProviderName**

The service provider name placed in the service_descriptor in the Service Description Table. Maximum length is 256 characters.

**Type**: string
**Required**: False

**outputSdt**

**Type**: string
**Required**: False

**DvbSubDestinationSettings**

**xPosition**

Specifies the horizontal position of the caption relative to the left side of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the left of the output. If no explicit x_position is provided, the horizontal caption position will be determined by the alignment parameter. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

**Type**: integer
**Required**: False
**Minimum**: 0
**Maximum**: 2147483647

**backgroundColor**

**Type**: string
**Required**: False

**teletextSpacing**

**Type**: string
**Required**: False

**yPosition**

Specifies the vertical position of the caption relative to the top of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the top of the output. If no explicit y_position is provided, the caption will be positioned towards the bottom of the output. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

**Type**: integer
**Required**: False
**Minimum**: 0
**Maximum**: 2147483647
backgroundOpacity

Specifies the opacity of the background rectangle. 255 is opaque; 0 is transparent. Leaving this parameter blank is equivalent to setting it to 0 (transparent). All burn-in and DVB-Sub font settings must match.

Type: integer  
Required: False  
Minimum: 0  
Maximum: 255

fontOpacity

Specifies the opacity of the burned-in captions. 255 is opaque; 0 is transparent. All burn-in and DVB-Sub font settings must match.

Type: integer  
Required: True  
Minimum: 0  
Maximum: 255

fontResolution

Font resolution in DPI (dots per inch); default is 96 dpi. All burn-in and DVB-Sub font settings must match.

Type: integer  
Required: False  
Minimum: 96  
Maximum: 600

shadowOpacity

Specifies the opacity of the shadow. 255 is opaque; 0 is transparent. Leaving this parameter blank is equivalent to setting it to 0 (transparent). All burn-in and DVB-Sub font settings must match.

Type: integer  
Required: False  
Minimum: 0  
Maximum: 255

shadowYOffset

Specifies the vertical offset of the shadow relative to the captions in pixels. A value of -2 would result in a shadow offset 2 pixels above the text. All burn-in and DVB-Sub font settings must match.

Type: integer  
Required: False  
Minimum: -2147483648  
Maximum: 2147483647

outlineSize

Specifies font outline size in pixels. This option is not valid for source captions that are either 608/ embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.
Type: integer
Required: True
Minimum: 0
Maximum: 10

outlineColor
Type: string
Required: True

fontSize
A positive integer indicates the exact font size in points. Set to 0 for automatic font size selection. All burn-in and DVB-Sub font settings must match.
Type: integer
Required: False
Minimum: 0
Maximum: 96

shadowXOffset
Specifies the horizontal offset of the shadow relative to the captions in pixels. A value of -2 would result in a shadow offset 2 pixels to the left. All burn-in and DVB-Sub font settings must match.
Type: integer
Required: False
Minimum: -2147483648
Maximum: 2147483647

alignment
Type: string
Required: True

shadowColor
Type: string
Required: False

fontColor
Type: string
Required: False

DvbSubSourceSettings

id
When using DVB-Sub with Burn-In or SMPTE-TT, use this PID for the source content. Unused for DVB-Sub passthrough. All DVB-Sub content is passed through, regardless of selectors.
Type: integer
**Required:** False
**Minimum:** 1
**Maximum:** 2147483647

**DvbSubtitleAlignment (enum)**

If no explicit x_position or y_position is provided, setting alignment to centered will place the captions at the bottom center of the output. Similarly, setting a left alignment will align captions to the bottom left of the output. If x and y positions are given in conjunction with the alignment parameter, the font will be justified (either left or centered) relative to those coordinates. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- CENTERED
- LEFT

**DvbSubtitleBackgroundColor (enum)**

Specifies the color of the rectangle behind the captions. All burn-in and DVB-Sub font settings must match.

- NONE
- BLACK
- WHITE

**DvbSubtitleFontColor (enum)**

Specifies the color of the burned-in captions. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- WHITE
- BLACK
- YELLOW
- RED
- GREEN
- BLUE

**DvbSubtitleOutlineColor (enum)**

Specifies font outline color. This option is not valid for source captions that are either 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- BLACK
- WHITE
- YELLOW
- RED
- GREEN
- BLUE

**DvbSubtitleShadowColor (enum)**

Specifies the color of the shadow cast by the captions. All burn-in and DVB-Sub font settings must match.
DvbSubtitleTeletextSpacing (enum)

Only applies to jobs with input captions in Teletext or STL formats. Specify whether the spacing between letters in your captions is set by the captions grid or varies depending on letter width. Choose fixed grid to conform to the spacing specified in the captions file more accurately. Choose proportional to make the text easier to read if the captions are closed caption.

- FIXED_GRID
- PROPORTIONAL

DvbTdtSettings

tdtInterval

The number of milliseconds between instances of this table in the output transport stream.

- **Type**: integer
- **Required**: True
- **Minimum**: 1000
- **Maximum**: 30000

Eac3AttenuationControl (enum)

If set to ATTENUATE_3_DB, applies a 3 dB attenuation to the surround channels. Only used for 3/2 coding mode.

- ATTENUATE_3_DB
- NONE

Eac3BitstreamMode (enum)

Specifies the "Bitstream Mode" (bsmod) for the emitted E-AC-3 stream. See ATSC A/52-2012 (Annex E) for background on these values.

- COMPLETE_MAIN
- COMMENTARY
- EMERGENCY
- HEARING_IMPAIRED
- VISUALLY_IMPAIRED

Eac3CodingMode (enum)

Dolby Digital Plus coding mode. Determines number of channels.

- CODING_MODE_1_0
- CODING_MODE_2_0
- CODING_MODE_3_2

Eac3DcFilter (enum)

Activates a DC highpass filter for all input channels.
ENABLED
DISABLED

Eac3DynamicRangeCompressionLine (enum)
Enables Dynamic Range Compression that restricts the absolute peak level for a signal.

NONE
FILM_STANDARD
FILM_LIGHT
MUSIC_STANDARD
MUSIC_LIGHT
SPEECH

Eac3DynamicRangeCompressionRf (enum)
Enables Heavy Dynamic Range Compression, ensures that the instantaneous signal peaks do not exceed specified levels.

NONE
FILM_STANDARD
FILM_LIGHT
MUSIC_STANDARD
MUSIC_LIGHT
SPEECH

Eac3LfeControl (enum)
When encoding 3/2 audio, controls whether the LFE channel is enabled

LFE
NO_LFE

Eac3LfeFilter (enum)
Applies a 120Hz lowpass filter to the LFE channel prior to encoding. Only valid with 3_2_LFE coding mode.

ENABLED
DISABLED

Eac3MetadataControl (enum)
When set to FOLLOW_INPUT, encoder metadata will be sourced from the DD, DD+, or DolbyE decoder that supplied this audio data. If audio was not supplied from one of these streams, then the static metadata settings will be used.

FOLLOW_INPUT
USE_CONFIGURED

Eac3PassthroughControl (enum)
When set to WHEN_POSSIBLE, input DD+ audio will be passed through if it is present on the input. This detection is dynamic over the life of the transcode. Inputs that alternate between DD+ and non-DD+ content will have a consistent DD+ output as the system alternates between passthrough and encoding.
Eac3PhaseControl (enum)
Controls the amount of phase-shift applied to the surround channels. Only used for 3/2 coding mode.

- SHIFT_90_DEGREES
- NO_SHIFT

Eac3Settings

dialnorm
Sets the dialnorm for the output. If blank and input audio is Dolby Digital Plus, dialnorm will be passed through.

- Type: integer
- Required: False
- Minimum: 1
- Maximum: 31

passthroughControl

- Type: string
- Required: False

metadataControl

- Type: string
- Required: False

bitrate
Average bitrate in bits/second. Valid bitrates depend on the coding mode.

- Type: integer
- Required: False
- Minimum: 64000
- Maximum: 640000

dynamicRangeCompressionRf

- Type: string
- Required: False

sampleRate
Sample rate in hertz. Sample rate is always 48000.

- Type: integer
- Required: False
- Minimum: 48000
Maximum: 48000

**surroundExMode**

Type: string  
Required: False

**ltRtSurroundMixLevel**

Left total/Right total surround mix level. Only used for 3/2 coding mode. Valid values: -1.5 -3.0 -4.5 -6.0 -60

Type: number  
Required: False  
Format: float  
Minimum: -60.0  
Maximum: -1.5

**dynamicRangeCompressionLine**

Type: string  
Required: False

**lfeControl**

Type: string  
Required: False

**codingMode**

Type: string  
Required: False

**surroundMode**

Type: string  
Required: False

**attenuationControl**

Type: string  
Required: False

**lfeFilter**

Type: string  
Required: False

**phaseControl**

Type: string  
Required: False
**ltRtCenterMixLevel**

Left total/Right total center mix level. Only used for 3/2 coding mode. Valid values: 3.0, 1.5, 0.0, -1.5, -3.0, -4.5, -6.0, -60

- **Type:** number
  - **Required:** False
  - **Format:** float
  - **Minimum:** -60.0
  - **Maximum:** 3.0

**dcFilter**

- **Type:** string
  - **Required:** False

**bitstreamMode**

- **Type:** string
  - **Required:** False

**stereoDownmix**

- **Type:** string
  - **Required:** False

**loRoSurroundMixLevel**

Left only/Right only surround mix level. Only used for 3/2 coding mode. Valid values: -1.5, -3.0, -4.5, -6.0, -60

- **Type:** number
  - **Required:** False
  - **Format:** float
  - **Minimum:** -60.0
  - **Maximum:** -1.5

**loRoCenterMixLevel**

Left only/Right only center mix level. Only used for 3/2 coding mode. Valid values: 3.0, 1.5, 0.0, -1.5, -3.0, -4.5, -6.0, -60

- **Type:** number
  - **Required:** False
  - **Format:** float
  - **Minimum:** -60.0
  - **Maximum:** 3.0

**Eac3StereoDownmix (enum)**

Stereo downmix preference. Only used for 3/2 coding mode.

- **NOT_INDICATED**
- **LO_RO**
LT_RT
DPL2

**Eac3SurroundExMode (enum)**

When encoding 3/2 audio, sets whether an extra center back surround channel is matrix encoded into the left and right surround channels.

- NOT_INDICATED
- ENABLED
- DISABLED

**Eac3SurroundMode (enum)**

When encoding 2/0 audio, sets whether Dolby Surround is matrix encoded into the two channels.

- NOT_INDICATED
- ENABLED
- DISABLED

**EmbeddedConvert608To708 (enum)**

When set to UPCONVERT, 608 data is both passed through via the "608 compatibility bytes" fields of the 708 wrapper as well as translated into 708. 708 data present in the source content will be discarded.

- UPCONVERT
- DISABLED

**EmbeddedSourceSettings**

**source608ChannelNumber**

Specifies the 608/708 channel number within the video track from which to extract captions. Unused for passthrough.

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 4

**convert608To708**

- **Type**: string
- **Required**: False

**source608TrackNumber**

Specifies the video track index used for extracting captions. The system only supports one input video track, so this should always be set to '1'.

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 1
**ExceptionBody**

- **message**
  - Type: string
  - Required: False

**F4vMoovPlacement (enum)**

If set to PROGRESSIVE_DOWNLOAD, the MOOV atom is relocated to the beginning of the archive as required for progressive downloading. Otherwise it is placed normally at the end.

- PROGRESSIVE_DOWNLOAD
- NORMAL

**F4vSettings**

- **moovPlacement**
  - Type: string
  - Required: False

**FileGroupSettings**

- **destination**

  Use Destination (Destination) to specify the S3 output location and the output filename base. Destination accepts format identifiers. If you do not specify the base filename in the URI, the service will use the filename of the input file. If your job has multiple inputs, the service uses the filename of the first input file.

  - Type: string
  - Required: False
  - Pattern: ^s3:/\/

**FileSourceConvert608To708 (enum)**

If set to UPCONVERT, 608 caption data is both passed through via the "608 compatibility bytes" fields of the 708 wrapper as well as translated into 708. 708 data present in the source content will be discarded.

- UPCONVERT
- DISABLED

**FileSourceSettings**

- **timeDelta**

  Specifies a time delta in seconds to offset the captions from the source file.

  - Type: integer
  - Required: False
  - Minimum: -2147483648
  - Maximum: 2147483647
**convert608To708**

*Type:* string  
*Required:* False

**sourceFile**

External caption file used for loading captions. Accepted file extensions are 'scc', 'ttml', 'dfxp', 'stl', 'srt', and 'smi'.

*Type:* string  
*Required:* True

*Pattern:* ^s3://(.*?).(scc|SCC|ttml|TTML|dfxp|DFXP|stl|STL|srt|SRT|smi|SMI)$

**FrameCaptureSettings**

**framerateDenominator**

Frame capture will encode the first frame of the output stream, then one frame every framerateDenominator/framerateNumerator seconds. For example, settings of framerateNumerator = 1 and framerateDenominator = 3 (a rate of 1/3 frame per second) will capture the first frame, then 1 frame every 3s. Files will be named as filename.n.jpg where n is the 0-based sequence number of each Capture.

*Type:* integer  
*Required:* False

*Minimum:* 1  
*Maximum:* 2147483647

**maxCaptures**

Maximum number of captures (encoded jpg output files).

*Type:* integer  
*Required:* False

*Minimum:* 1  
*Maximum:* 10000000

**framerateNumerator**

Frame capture will encode the first frame of the output stream, then one frame every framerateDenominator/framerateNumerator seconds. For example, settings of framerateNumerator = 1 and framerateDenominator = 3 (a rate of 1/3 frame per second) will capture the first frame, then 1 frame every 3s. Files will be named as filename.NNNNNNN.jpg where N is the 0-based frame sequence number zero padded to 7 decimal places.

*Type:* integer  
*Required:* False

*Minimum:* 1  
*Maximum:* 2147483647

**quality**

JPEG Quality - a higher value equals higher quality.
Type: integer
Required: False
Minimum: 1
Maximum: 100

**GetJobRequest**

id

the job ID of the job.

Type: string
Required: False

**GetJobResponse**

job

Type: Job (p. 592)
Required: False

**H264AdaptiveQuantization (enum)**

Adaptive quantization. Allows intra-frame quantizers to vary to improve visual quality.

OFF
LOW
MEDIUM
HIGH
HIGHER
MAX

**H264CodecLevel (enum)**

H.264 Level.

AUTO
LEVEL_1
LEVEL_1_1
LEVEL_1_2
LEVEL_1_3
LEVEL_2
LEVEL_2_1
LEVEL_2_2
LEVEL_3
LEVEL_3_1
LEVEL_3_2
LEVEL_4
LEVEL_4_1
LEVEL_4_2
LEVEL_5
LEVEL_5_1
LEVEL_5_2

**H264CodecProfile (enum)**

H.264 Profile. High 4:2:2 and 10-bit profiles are only available with the AVC-I License.

- BASELINE
- HIGH
- HIGH_10BIT
- HIGH_422
- HIGH_422_10BIT
- MAIN

**H264EntropyEncoding (enum)**

Entropy encoding mode. Use CABAC (must be in Main or High profile) or CAVLC.

- CABAC
- CAVLC

**H264FieldEncoding (enum)**

Choosing FORCE_FIELD disables PAFF encoding for interlaced outputs.

- PAFF
- FORCE_FIELD

**H264FlickerAdaptiveQuantization (enum)**

Adjust quantization within each frame to reduce flicker or 'pop' on I-frames.

- DISABLED
- ENABLED

**H264FramerateControl (enum)**

If you are using the console, use the Framerate setting to specify the framerate for this output. If you want to keep the same framerate as the input video, choose Follow source. If you want to do framerate conversion, choose a framerate from the dropdown list or choose Custom. The framerates shown in the dropdown list are decimal approximations of fractions. If you choose Custom, specify your framerate as a fraction. If you are creating your transcoding job specification as a JSON file without the console, use FramerateControl to specify which value the service uses for the framerate for this output. Choose INITIALIZE_FROM_SOURCE if you want the service to use the framerate from the input. Choose SPECIFIED if you want the service to use the framerate you specify in the settings FramerateNumerator and FramerateDenominator.

- INITIALIZE_FROM_SOURCE
- SPECIFIED

**H264FramerateConversionAlgorithm (enum)**

When set to INTERPOLATE, produces smoother motion during framerate conversion.

- DUPLICATE_DROP
- INTERPOLATE
**H264GopBReference (enum)**

If enable, use reference B frames for GOP structures that have B frames > 1.

- DISABLED
- ENABLED

**H264GopSizeUnits (enum)**

Indicates if the GOP Size in H264 is specified in frames or seconds. If seconds the system will convert the GOP Size into a frame count at run time.

- FRAMES
- SECONDS

**H264InterlaceMode (enum)**

Use Interlace mode (InterlaceMode) to choose the scan line type for the output. * Top Field First (TOP_FIELD) and Bottom Field First (BOTTOM_FIELD) produce interlaced output with the entire output having the same field polarity (top or bottom first). * Follow, Default Top (FOLLOW_TOP_FIELD) and Follow, Default Bottom (FOLLOW_BOTTOM_FIELD) use the same field polarity as the source. Therefore, behavior depends on the input scan type, as follows. - If the source is interlaced, the output will be interlaced with the same polarity as the source (it will follow the source). The output could therefore be a mix of "top field first" and "bottom field first". - If the source is progressive, the output will be interlaced with "top field first" or "bottom field first" polarity, depending on which of the Follow options you chose.

- PROGRESSIVE
- TOP_FIELD
- BOTTOM_FIELD
- FOLLOW_TOP_FIELD
- FOLLOW_BOTTOM_FIELD

**H264ParControl (enum)**

Using the API, enable ParFollowSource if you want the service to use the pixel aspect ratio from the input. Using the console, do this by choosing Follow source for Pixel aspect ratio.

- INITIALIZE_FROM_SOURCE
- SPECIFIED

**H264QualityTuningLevel (enum)**

Use Quality tuning level (H264QualityTuningLevel) to specify whether to use fast single-pass, high-quality singlepass, or high-quality multipass video encoding.

- SINGLE_PASS
- SINGLE_PASS_HQ
- MULTI_PASS_HQ

**H264RateControlMode (enum)**

Use this setting to specify whether this output has a variable bitrate (VBR) or constant bitrate (CBR).

- VBR
- CBR
**H264RepeatPps (enum)**
Places a PPS header on each encoded picture, even if repeated.

- DISABLED
- ENABLED

**H264SceneChangeDetect (enum)**
Scene change detection (inserts I-frames on scene changes).

- DISABLED
- ENABLED

**H264Settings**

**slices**
Number of slices per picture. Must be less than or equal to the number of macroblock rows for progressive pictures, and less than or equal to half the number of macroblock rows for interlaced pictures.

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 32

**minIInterval**
Enforces separation between repeated (cadence) I-frames and I-frames inserted by Scene Change Detection. If a scene change I-frame is within I-interval frames of a cadence I-frame, the GOP is shrunk and/or stretched to the scene change I-frame. GOP stretch requires enabling lookahead as well as setting I-interval. The normal cadence resumes for the next GOP. This setting is only used when Scene Change Detect is enabled. Note: Maximum GOP stretch = GOP size + Min-I-interval - 1

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 30

**parNumerator**
Pixel Aspect Ratio numerator.

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 2147483647

**flickerAdaptiveQuantization**

- **Type**: string
- **Required**: False
gopSizeUnits
  Type: string
  Required: False

hrdBufferSize
Size of buffer (HRD buffer model) in bits. For example, enter five megabits as 5000000.
  Type: integer
  Required: False
  Minimum: 0
  Maximum: 1152000000

qualityTuningLevel
  Type: string
  Required: False

maxBitrate
Maximum bitrate in bits/second. For example, enter five megabits per second as 5000000.
  Type: integer
  Required: False
  Minimum: 1000
  Maximum: 1152000000

bitrate
Average bitrate in bits/second. Required for VBR and CBR. For MS Smooth outputs, bitrates must be unique when rounded down to the nearest multiple of 1000.
  Type: integer
  Required: False
  Minimum: 1000
  Maximum: 1152000000

spatialAdaptiveQuantization
  Type: string
  Required: False

slowPal
  Type: string
  Required: False

codecProfile
  Type: string
  Required: False
unregisteredSeiTimecode

Type: string
Required: False

softness

Softness. Selects quantizer matrix, larger values reduce high-frequency content in the encoded image.

Type: integer
Required: False
Minimum: 0
Maximum: 128

framerateControl

Type: string
Required: False

telecine

Type: string
Required: False

codecLevel

Type: string
Required: False

framerateConversionAlgorithm

Type: string
Required: False

numberReferenceFrames

Number of reference frames to use. The encoder may use more than requested if using B-frames and/or interlaced encoding.

Type: integer
Required: False
Minimum: 1
Maximum: 6

temporalAdaptiveQuantization

Type: string
Required: False

repeatPps

Type: string
Required: False
**hrdBufferInitialFillPercentage**

Percentage of the buffer that should initially be filled (HRD buffer model).

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 100

**framerateNumerator**

Framerate numerator - framerate is a fraction, e.g. 24000 / 1001 = 23.976 fps.

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 2147483647

**gopClosedCadence**

Frequency of closed GOPs. In streaming applications, it is recommended that this be set to 1 so a decoder joining mid-stream will receive an IDR frame as quickly as possible. Setting this value to 0 will break output segmenting.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 2147483647

**numberBFramesBetweenReferenceFrames**

Number of B-frames between reference frames.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 7

**framerateDenominator**

When you use the API for transcode jobs that use framerate conversion, specify the framerate as a fraction. For example, 24000 / 1001 = 23.976 fps. Use FramerateDenominator to specify the denominator of this fraction. In this example, use 1001 for the value of FramerateDenominator. When you use the console for transcode jobs that use framerate conversion, provide the value as a decimal number for Framerate. In this example, specify 23.976.

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 2147483647

**entropyEncoding**

- **Type**: string
- **Required**: False
**fieldEncoding**

Type: string  
Required: False

**adaptiveQuantization**

Type: string  
Required: False

**interlaceMode**

Type: string  
Required: False

**gopSize**

GOP Length (keyframe interval) in frames or seconds. Must be greater than zero.

Type: number  
Required: False  
Format: float  
Minimum: 0.0

**gopBReference**

Type: string  
Required: False

**syntax**

Type: string  
Required: False

**parDenominator**

Pixel Aspect Ratio denominator.

Type: integer  
Required: False  
Minimum: 1  
Maximum: 2147483647

**sceneChangeDetect**

Type: string  
Required: False

**parControl**

Type: string
Required: False

rateControlMode

Type: string
Required: False

H264SlowPal (enum)

Enables Slow PAL rate conversion. 23.976fps and 24fps input is relabeled as 25fps, and audio is sped up correspondingly.

DISABLED
ENABLED

H264SpatialAdaptiveQuantization (enum)

Adjust quantization within each frame based on spatial variation of content complexity.

DISABLED
ENABLED

H264Syntax (enum)

Produces a bitstream compliant with SMPTE RP-2027.

DEFAULT
RP2027

H264Telecine (enum)

This field applies only if the Streams > Advanced > Framerate (framerate) field is set to 29.970. This field works with the Streams > Advanced > Preprocessors > Deinterlacer field (deinterlace_mode) and the Streams > Advanced > Interlaced Mode field (interlace_mode) to identify the scan type for the output: Progressive, Interlaced, Hard Telecine or Soft Telecine. - Hard: produces 29.97i output from 23.976 input. - Soft: produces 23.976; the player converts this output to 29.97i.

NONE
SOFT
HARD

H264TemporalAdaptiveQuantization (enum)

Adjust quantization within each frame based on temporal variation of content complexity.

DISABLED
ENABLED

H264UnregisteredSeiTimecode (enum)

Inserts timecode for each frame as 4 bytes of an unregistered SEI message.

DISABLED
ENABLED
**H265AdaptiveQuantization (enum)**

Adaptive quantization. Allows intra-frame quantizers to vary to improve visual quality.

- OFF
- LOW
- MEDIUM
- HIGH
- HIGHER
- MAX

**H265AlternateTransferFunctionSei (enum)**

Enables Alternate Transfer Function SEI message for outputs using Hybrid Log Gamma (HLG) Electro-Optical Transfer Function (EOTF).

- DISABLED
- ENABLED

**H265CodecLevel (enum)**

H.265 Level.

- AUTO
- LEVEL_1
- LEVEL_2
- LEVEL_2_1
- LEVEL_3
- LEVEL_3_1
- LEVEL_4
- LEVEL_4_1
- LEVEL_5
- LEVEL_5_1
- LEVEL_5_2
- LEVEL_6
- LEVEL_6_1
- LEVEL_6_2

**H265CodecProfile (enum)**

Represents the Profile and Tier, per the HEVC (H.265) specification. Selections are grouped as [Profile] / [Tier], so "Main/High" represents Main Profile with High Tier. 4:2:2 profiles are only available with the HEVC 4:2:2 License.

- MAIN_MAIN
- MAIN_HIGH
- MAIN10_MAIN
- MAIN10_HIGH
- MAIN_422_8BIT_MAIN
- MAIN_422_8BIT_HIGH
- MAIN_422_10BIT_MAIN
- MAIN_422_10BIT_HIGH

**H265FlickerAdaptiveQuantization (enum)**

Adjust quantization within each frame to reduce flicker or 'pop' on I-frames.
DISABLED
ENABLED

H265FramerateControl (enum)

If you are using the console, use the Framerate setting to specify the framerate for this output. If you want to keep the same framerate as the input video, choose Follow source. If you want to do framerate conversion, choose a framerate from the dropdown list or choose Custom. The framerates shown in the dropdown list are decimal approximations of fractions. If you choose Custom, specify your framerate as a fraction. If you are creating your transcoding job specification as a JSON file without the console, use FramerateControl to specify which value the service uses for the framerate for this output. Choose INITIALIZE_FROM_SOURCE if you want the service to use the framerate from the input. Choose SPECIFIED if you want the service to use the framerate you specify in the settings FramerateNumerator and FramerateDenominator.

INITIALIZE_FROM_SOURCE
SPECIFIED

H265FramerateConversionAlgorithm (enum)

When set to INTERPOLATE, produces smoother motion during framerate conversion.

DUPLICATE_DROP
INTERPOLATE

H265GopBReference (enum)

If enabled, use reference B frames for GOP structures that have B frames > 1.

DISABLED
ENABLED

H265GopSizeUnits (enum)

Indicates if the GOP Size in H265 is specified in frames or seconds. If seconds the system will convert the GOP Size into a frame count at run time.

FRAMES
SECONDS

H265InterlaceMode (enum)

Use Interlaced mode (InterlaceMode) to choose the scan line type for the output. * Top Field First (TOP_FIELD) and Bottom Field First (BOTTOM_FIELD) produce interlaced output with the entire output having the same field polarity (top or bottom first). * Follow, Default Top (FOLLOW_TOP_FIELD) and Follow, Default Bottom (FOLLOW_BOTTOM_FIELD) use the same field polarity as the source. Therefore, behavior depends on the input scan type. - If the source is interlaced, the output will be interlaced with the same polarity as the source (it will follow the source). The output could therefore be a mix of "top field first" and "bottom field first". * If the source is progressive, the output will be interlaced with "top field first" or "bottom field first" polarity, depending on which of the Follow options you chose.

PROGRESSIVE
TOP_FIELD
BOTTOM_FIELD
FOLLOW_TOP_FIELD
H265ParControl (enum)
Using the API, enable ParFollowSource if you want the service to use the pixel aspect ratio from the input. Using the console, do this by choosing Follow source for Pixel aspect ratio.

INITIALIZE_FROM_SOURCE
SPECIFIED

H265QualityTuningLevel (enum)
Use Quality tuning level (H265QualityTuningLevel) to specify whether to use fast single-pass, high-quality singlepass, or high-quality multipass video encoding.

SINGLE_PASS
SINGLE_PASS_HQ
MULTI_PASS_HQ

H265RateControlMode (enum)
Use this setting to specify whether this output has a variable bitrate (VBR) or constant bitrate (CBR).

VBR
CBR

H265SampleAdaptiveOffsetFilterMode (enum)
Specify Sample Adaptive Offset (SAO) filter strength. Adaptive mode dynamically selects best strength based on content.

DEFAULT
ADAPTIVE
OFF

H265SceneChangeDetect (enum)
Scene change detection (inserts I-frames on scene changes).

DISABLED
ENABLED

H265Settings
slices
Number of slices per picture. Must be less than or equal to the number of macroblock rows for progressive pictures, and less than or equal to half the number of macroblock rows for interlaced pictures.

Type: integer
Required: False
Minimum: 1
**Properties**

**minInterval**
Enforces separation between repeated (cadence) I-frames and I-frames inserted by Scene Change Detection. If a scene change I-frame is within I-interval frames of a cadence I-frame, the GOP is shrunk and/or stretched to the scene change I-frame. GOP stretch requires enabling lookahead as well as setting I-interval. The normal cadence resumes for the next GOP. This setting is only used when Scene Change Detect is enabled. Note: Maximum GOP stretch = GOP size + Min-I-interval - 1

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 30

**parNumerator**
Pixel Aspect Ratio numerator.

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 2147483647

**flickerAdaptiveQuantization**

- **Type**: string
- **Required**: False

**gopSizeUnits**

- **Type**: string
- **Required**: False

**hrdBufferSize**
Size of buffer (HRD buffer model) in bits. For example, enter five megabits as 5000000.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 1466400000

**qualityTuningLevel**

- **Type**: string
- **Required**: False

**maxBitrate**
Maximum bitrate in bits/second.

- **Type**: integer
- **Required**: False
Properties

Minimum: 1000
Maximum: 1466400000

**bitrate**

Average bitrate in bits/second. Required for VBR and CBR. For MS Smooth outputs, bitrates must be unique when rounded down to the nearest multiple of 1000.

- **Type:** integer
- **Required:** False
- **Minimum:** 1000
- **Maximum:** 1466400000

**spatialAdaptiveQuantization**

- **Type:** string
- **Required:** False

**sampleAdaptiveOffsetFilterMode**

- **Type:** string
- **Required:** False

**temporalIds**

- **Type:** string
- **Required:** False

**slowPal**

- **Type:** string
- **Required:** False

**tiles**

- **Type:** string
- **Required:** False

**writeMp4PackagingType**

- **Type:** string
- **Required:** False

**codecProfile**

- **Type:** string
- **Required:** False

**alternateTransferFunctionSei**

- **Type:** string
- **Required:** False
unregisteredSeiTimecode
  Type: string
  Required: False

framerateControl
  Type: string
  Required: False

telecine
  Type: string
  Required: False

codecLevel
  Type: string
  Required: False

framerateConversionAlgorithm
  Type: string
  Required: False

numberReferenceFrames
  Number of reference frames to use. The encoder may use more than requested if using B-frames and/or
  interlaced encoding.
  Type: integer
  Required: False
  Minimum: 1
  Maximum: 6

temporalAdaptiveQuantization
  Type: string
  Required: False

hrdBufferInitialFillPercentage
  Percentage of the buffer that should initially be filled (HRD buffer model).
  Type: integer
  Required: False
  Minimum: 0
  Maximum: 100

framerateNumerator
  Framerate numerator - framerate is a fraction, e.g. 24000 / 1001 = 23.976 fps.
**Type**: integer  
**Required**: False  
**Minimum**: 1  
**Maximum**: 2147483647

**gopClosedCadence**

Frequency of closed GOPs. In streaming applications, it is recommended that this be set to 1 so a decoder joining mid-stream will receive an IDR frame as quickly as possible. Setting this value to 0 will break output segmenting.

**Type**: integer  
**Required**: False  
**Minimum**: 0  
**Maximum**: 2147483647

**numberBFramesBetweenReferenceFrames**

Number of B-frames between reference frames.

**Type**: integer  
**Required**: False  
**Minimum**: 0  
**Maximum**: 7

**framerateDenominator**

Framerate denominator.

**Type**: integer  
**Required**: False  
**Minimum**: 1  
**Maximum**: 2147483647

**adaptiveQuantization**

**Type**: string  
**Required**: False

**interlaceMode**

**Type**: string  
**Required**: False

**gopSize**

GOP Length (keyframe interval) in frames or seconds. Must be greater than zero.

**Type**: number  
**Required**: False  
**Format**: float  
**Minimum**: 0.0
**gopBReference**

Type: string  
Required: False

**parDenominator**

Pixel Aspect Ratio denominator.

Type: integer  
Required: False  
Minimum: 1  
Maximum: 2147483647

**sceneChangeDetect**

Type: string  
Required: False

**parControl**

Type: string  
Required: False

**rateControlMode**

Type: string  
Required: False

**H265SlowPal (enum)**

Enables Slow PAL rate conversion. 23.976fps and 24fps input is relabeled as 25fps, and audio is sped up correspondingly.

DISABLED  
ENABLED

**H265SpatialAdaptiveQuantization (enum)**

Adjust quantization within each frame based on spatial variation of content complexity.

DISABLED  
ENABLED

**H265Telecine (enum)**

This field applies only if the Streams > Advanced > Framerate (framerate) field is set to 29.970. This field works with the Streams > Advanced > Preprocessors > Deinterlacer field (deinterlace_mode) and the Streams > Advanced > Interlaced Mode field (interlace_mode) to identify the scan type for the output: Progressive, Interlaced, Hard Telecine or Soft Telecine. - Hard: produces 29.97i output from 23.976 input. - Soft: produces 23.976; the player converts this output to 29.97i.

NONE  
SOFT
HARD

H265TemporalAdaptiveQuantization (enum)
Adjust quantization within each frame based on temporal variation of content complexity.

- DISABLED
- ENABLED

H265TemporalIds (enum)
Enables temporal layer identifiers in the encoded bitstream. Up to 3 layers are supported depending on GOP structure: I- and P-frames form one layer, reference B-frames can form a second layer and non-reference b-frames can form a third layer. Decoders can optionally decode only the lower temporal layers to generate a lower frame rate output. For example, given a bitstream with temporal IDs and with b-frames = 1 (i.e. IbPbPb display order), a decoder could decode all the frames for full frame rate output or only the I and P frames (lowest temporal layer) for a half frame rate output.

- DISABLED
- ENABLED

H265Tiles (enum)
Enable use of tiles, allowing horizontal as well as vertical subdivision of the encoded pictures.

- DISABLED
- ENABLED

H265UnregisteredSeiTimecode (enum)
Inserts timecode for each frame as 4 bytes of an unregistered SEI message.

- DISABLED
- ENABLED

H265WriteMp4PackagingType (enum)
If HVC1, output that is H.265 will be marked as HVC1 and adhere to the ISO-IECJTC1-SC29_N13798_Text_ISOIEC_FDIS_14496-15_3rd_E spec which states that parameter set NAL units will be stored in the sample headers but not in the samples directly. If HEV1, then H.265 will be marked as HEV1 and parameter set NAL units will be written into the samples.

- HVC1
- HEV1

Hdr10Metadata

redPrimaryY
HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

- Type: integer

576
Required: False
Minimum: 0
Maximum: 50000

greenPrimaryY

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

Type: integer
Required: False
Minimum: 0
Maximum: 50000

whitePointX

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

Type: integer
Required: False
Minimum: 0
Maximum: 50000

maxLuminance

Nominal maximum mastering display luminance in units of 0.0001 candelas per square meter.

Type: integer
Required: False
Minimum: 0
Maximum: 2147483647

greenPrimaryX

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

Type: integer
Required: False
Minimum: 0
Maximum: 50000

whitePointY

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

Type: integer
Required: False
**redPrimaryX**

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 50000

**bluePrimaryX**

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 50000

**bluePrimaryY**

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 50000

**maxFrameAverageLightLevel**

Maximum average light level of any frame in the coded video sequence, in units of candelas per square meter.

- **Type**: integer
- **Required**: True
- **Minimum**: 0
- **Maximum**: 65535

**maxContentLightLevel**

Maximum light level among all samples in the coded video sequence, in units of candelas per square meter.

- **Type**: integer
- **Required**: True
- **Minimum**: 0
**Properties**

**Minimum**: 65535

**minLuminance**

Nominal minimum mastering display luminance in units of 0.0001 candelas per square meter.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 2147483647

**HlsMarkers (enum)**

- **ELEMENTAL**
- **ELEMENTAL_SCTE35**

**HlsAudioTrackType (enum)**

Four types of audio-only tracks are supported: Audio-Only Variant Stream. The client can play back this audio-only stream instead of video in low-baudwidth scenarios. Represented as an EXT-X-STREAM-INF in the HLS manifest. Alternate Audio, Auto Select, Default Alternate rendition that the client should try to play back by default. Represented as an EXT-X-MEDIA in the HLS manifest with DEFAULT=YES, AUTOSELECT=YES Alternate Audio, Auto Select, Not Default Alternate rendition that the client may try to play back by default. Represented as an EXT-X-MEDIA in the HLS manifest with DEFAULT=NO, AUTOSELECT=YES Alternate Audio, not Auto Select Alternate rendition that the client will not try to play back by default. Represented as an EXT-X-MEDIA in the HLS manifest with DEFAULT=NO, AUTOSELECT=NO

- **ALTERNATE_AUDIO_AUTO_SELECT_DEFAULT**
- **ALTERNATE_AUDIO_AUTO_SELECT**
- **ALTERNATE_AUDIO_NOT_AUTO_SELECT**
- **AUDIO_ONLY_VARIANT_STREAM**

**HlsCaptionLanguageMapping**

**languageDescription**

Caption language description.

- **Type**: string
- **Required**: False

**captionChannel**

Caption channel.

- **Type**: integer
- **Required**: False
- **Minimum**: -2147483648
- **Maximum**: 2147483647

**languageCode**

- **Type**: string
**Properties**

**Required**: False

**customLanguageCode**

Specify the language for this caption channel, using the ISO 639-2 or ISO 639-3 three-letter language code.

**Type**: string

**Required**: False

**Pattern**: ^[A-Za-z]{3}$

**HlsCaptionLanguageSetting (enum)**

Applies only to 608 Embedded output captions. Insert: Include CLOSED-CAPTIONS lines in the manifest. Specify at least one language in the CC1 Language Code field. One CLOSED-CAPTION line is added for each Language Code you specify. Make sure to specify the languages in the order in which they appear in the original source (if the source is embedded format) or the order of the caption selectors (if the source is other than embedded). Otherwise, languages in the manifest will not match up properly with the output captions. None: Include CLOSED-CAPTIONS=None line in the manifest. Omit: Omit any CLOSED-CAPTIONS line from the manifest.

- INSERT
- OMIT
- NONE

**HlsClientCache (enum)**

When set to ENABLED, sets #EXT-X-ALLOW-CACHE:no tag, which prevents client from saving media segments for later replay.

- DISABLED
- ENABLED

**HlsCodecSpecification (enum)**

Specification to use (RFC-6381 or the default RFC-4281) during m3u8 playlist generation.

- RFC_6381
- RFC_4281

**HlsDirectoryStructure (enum)**

Indicates whether segments should be placed in subdirectories.

- SINGLE_DIRECTORY
- SUBDIRECTORY_PER_STREAM

**HlsEncryptionSettings**

**initializationVectorInManifest**

**Type**: string

**Required**: False
**constantInitializationVector**

This is a 128-bit, 16-byte hex value represented by a 32-character text string. If this parameter is not set then the Initialization Vector will follow the segment number by default.

- **Type:** string
- **Required:** False
- **Pattern:** `^[0-9a-fA-F]{32}$`

**staticKeyProvider**

- **Type:** StaticKeyProvider (p. 630)
- **Required:** False

**type**

- **Type:** string
- **Required:** True

**encryptionMethod**

- **Type:** string
- **Required:** False

**spekeKeyProvider**

- **Type:** SpekeKeyProvider (p. 630)
- **Required:** False

**HlsEncryptionType (enum)**

Encrypts the segments with the given encryption scheme. Leave blank to disable. Selecting 'Disabled' in the web interface also disables encryption.

- AES128
- SAMPLE_AES

**HlsGroupSettings**

**segmentsPerSubdirectory**

Number of segments to write to a subdirectory before starting a new one. directoryStructure must be SINGLE_DIRECTORY for this setting to have an effect.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 2147483647

**streamInfResolution**

- **Type:** string
Required: False

timestampDeltaMilliseconds
Provides an extra millisecond delta offset to fine tune the timestamps.

Type: integer
Required: False
Minimum: -2147483648
Maximum: 2147483647

outputSelection

Type: string
Required: False

captionLanguageMappings
Language to be used on Caption outputs

Type: Array of type HlsCaptionLanguageMapping (p. 579)
Required: False

clientCache

Type: string
Required: False

codecSpecification

Type: string
Required: False

destination
Use Destination (Destination) to specify the S3 output location and the output filename base. Destination accepts format identifiers. If you do not specify the base filename in the URI, the service will use the filename of the input file. If your job has multiple inputs, the service uses the filename of the first input file.

Type: string
Required: False
Pattern: ^s3://\/

timedMetadataId3Frame

Type: string
Required: False

segmentControl

Type: string
Required: False
timedMetadataId3Period

Timed Metadata interval in seconds.

Type: integer
Required: False
Minimum: -2147483648
Maximum: 2147483647

captionLanguageSetting

Type: string
Required: False

minSegmentLength

When set, Minimum Segment Size is enforced by looking ahead and back within the specified range for a nearby avail and extending the segment size if needed.

Type: integer
Required: True
Minimum: 0
Maximum: 2147483647

directoryStructure

Type: string
Required: False

programDateTime

Type: string
Required: False

baseUrl

A partial URI prefix that will be prepended to each output in the media .m3u8 file. Can be used if base manifest is delivered from a different URL than the main .m3u8 file.

Type: string
Required: False

encryption

DRM settings.

Type: HlsEncryptionSettings (p. 580)
Required: False

adMarkers

Choose one or more ad marker types to pass SCTE35 signals through to this group of Apple HLS outputs.
Type: Array of type string  
Required: False

**programDateTimePeriod**

Period of insertion of EXT-X-PROGRAM-DATE-TIME entry, in seconds.

Type: integer  
Required: False  
Minimum: 0  
Maximum: 3600

**manifestCompression**

Type: string  
Required: False

**segmentLength**

Length of MPEG-2 Transport Stream segments to create (in seconds). Note that segments will end on the next keyframe after this number of seconds, so actual segment length may be longer.

Type: integer  
Required: True  
Minimum: 1  
Maximum: 2147483647

**manifestDurationFormat**

Type: string  
Required: False

**HlsIFrameOnlyManifest (enum)**

When set to INCLUDE, writes I-Frame Only Manifest in addition to the HLS manifest

INCLUDE  
EXCLUDE

**HlsInitializationVectorInManifest (enum)**

The Initialization Vector is a 128-bit number used in conjunction with the key for encrypting blocks. If set to INCLUDE, Initialization Vector is listed in the manifest. Otherwise Initialization Vector is not in the manifest.

INCLUDE  
EXCLUDE

**HlsKeyProviderType (enum)**

Indicates which type of key provider is used for encryption.

SPEKE
**STATIC_KEY**

**HlsManifestCompression (enum)**

When set to GZIP, compresses HLS playlist.

- GZIP
- NONE

**HlsManifestDurationFormat (enum)**

Indicates whether the output manifest should use floating point values for segment duration.

- FLOATING_POINT
- INTEGER

**HlsOutputSelection (enum)**

Indicates whether the .m3u8 manifest file should be generated for this HLS output group.

- MANIFESTS_AND_SEGMENTS
- SEGMENTS_ONLY

**HlsProgramDateTime (enum)**

Includes or excludes EXT-X-PROGRAM-DATE-TIME tag in .m3u8 manifest files. The value is calculated as follows: either the program date and time are initialized using the input timecode source, or the time is initialized using the input timecode source and the date is initialized using the timestamp_offset.

- INCLUDE
- EXCLUDE

**HlsSegmentControl (enum)**

When set to SINGLE_FILE, emits program as a single media resource (.ts) file, uses #EXT-X-BYTERANGE tags to index segment for playback.

- SINGLE_FILE
- SEGMENTED_FILES

**HlsSettings**

**iFrameOnlyManifest**

- Type: string
- Required: False

**audioRenditionSets**

List all the audio groups that are used with the video output stream. Input all the audio GROUP-IDs that are associated to the video, separate by ";".

- Type: string
Properties

Required: False

audioTrackType

Type: string
Required: False

segmentModifier

String concatenated to end of segment filenames. Accepts "Format Identifiers":#format_identifier_parameters.

Type: string
Required: False

audioGroupId

Specifies the group to which the audio Rendition belongs.

Type: string
Required: False

HlsStreamInfResolution (enum)

Include or exclude RESOLUTION attribute for video in EXT-X-STREAM-INF tag of variant manifest.

INCLUDE
EXCLUDE

HlsTimedMetadataId3Frame (enum)

Indicates ID3 frame that has the timecode.

NONE
PRIV
TDRL

Id3Insertion

id3

Use ID3 tag (Id3) to provide a tag value in base64-encode format.

Type: string
Required: True
Pattern: ^[A-Za-z0-9+\-/\ ]+=\{0,2\}$

timecode

Provide a Timecode (TimeCode) in HH:MM:SS:FF or HH:MM:SS;FF format.

Type: string
Properties

**Required:** True  
**Format:** timecode  
**Pattern:** `^([01][0-9]|2[0-4]):[0-5][0-9]:[0-5][0-9][0-9];[0-9]{2}$`

**ImageInserter**

**insertableImages**

Image to insert. Must be 32 bit windows BMP, PNG, or TGA file. Must not be larger than the output frames.

- **Type:** Array of type `InsertableImage (p. 590)`  
- **Required:** True

**Input**

**audioSelectors**

Use Audio selectors (AudioSelectors) to specify a track or set of tracks from the input that you will use in your outputs. You can use mutiple Audio selectors per input.

- **Type:** object  
- **Required:** False

**audioSelectorGroups**

Specifies set of audio selectors within an input to combine. An input may have multiple audio selector groups. See "Audio Selector Group":#inputs-audio_selector_group for more information.

- **Type:** object  
- **Required:** False

**videoSelector**

- **Type:** `VideoSelector (p. 638)`  
- **Required:** False

**denoiseFilter**

- **Type:** string  
- **Required:** False

**filterEnable**

- **Type:** string  
- **Required:** False

**deblockFilter**

- **Type:** string  
- **Required:** False
Properties

**filterStrength**

Use Filter strength (FilterStrength) to adjust the magnitude the input filter settings (Deblock and Denoise). The range is -5 to 5. Default is 0.

- **Type:** integer
- **Required:** False
- **Minimum:** -5
- **Maximum:** 5

**programNumber**

Use Program (programNumber) to select a specific program from within a multi-program transport stream. Note that Quad 4K is not currently supported. Default is the first program within the transport stream. If the program you specify doesn’t exist, the transcoding service will use this default.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 2147483647

**timecodeSource**

- **Type:** string
- **Required:** False

**captionSelectors**

Use Captions selectors (CaptionSelectors) to specify the captions data from the input that you will use in your outputs. You can use multiple captions selectors per input.

- **Type:** object
- **Required:** False

**fileInput**

Use Input (fileInput) to define the source file used in the transcode job. There can be multiple inputs in a job. These inputs are concatenated, in the order they are specified in the job, to create the output.

- **Type:** string
- **Required:** True
- **Pattern:**

\[
\text{(s3://)([^/]+/)+([^/]+)([^/]*)(\[[mM]\[vV]\|[mM][pP][eE][gG]\|}\[aA]\[vV]\[iI]\|([mM][pP]\[4\]|(\[fF\]|ll\][Vv]\]|([mM]\[pP]\[tT]\|([mM][pP]\[gG]\|([mM]\[4\][Vv]\]|tT]\|([rR]\[pP]\|([fF]\[4\]|([mM]\[2\]|(tT]\|([sS]\|([tT]\|([sS]\|264|([hH]\|264|([mM]\[kK]\[vV]\|([mM]\[oO]\[vV]\|([mM]\[tT]\|([sS]\|([mM]\[2\]|(tT]\|([wW]\|([mM]\[vV]\|([aA]\[sS]\|([fF]\|([vV]\|([oO]\[bB]\|3|([gG]\|([fF]\|3\|[mM]\[xX]\|([fF]\|([dD]\|([iI]\|([vV]\|([xX]\|([vV]\|([iI]\|([dD]\|([rR]\|([aA]\|([wW]\|([dD]\|([vV]\|([gG]\|([xX]\|([fF]\|([mM]\|([vV]\|3\|[gG]\|([vV]\|([mM]\|([fF]\|([mM]\|3\|[uU]\|8\|[lL]\|([cC]\|([hH]\|([gG]\|([xX]\|([fF]\|([mM]\|([pP]\|([eE]\|([gG]\|2\|[mM]\|([xX]\|([fF]\|([mM]\|([pP]\|([eE]\|([gG]\|2\|[mM]\|([xX]\|([fF]\|([hH]\|([dD]\|([wW]\|([aA]\|([vV]\|([yY]\|4\|[mM]))))}\$)
\]

**inputClippings**

(InputClippings) contains sets of start and end times that together specify a portion of the input to be used in the outputs. If you provide only a start time, the clip will be the entire input from that point to the end. If you provide only an end time, it will be the entire input up to that point. When you specify
more than one input clip, the transcoding service creates the job outputs by stringing the clips together in the order you specify them.

**Type:** Array of type InputClipping (p. 589)

**Required:** False

**psiControl**

**Type:** string

**Required:** False

### InputClipping

**startTimecode**

Set Start timecode (StartTimecode) to the beginning of the portion of the input you are clipping. The frame corresponding to the Start timecode value is included in the clip. Start timecode or End timecode may be left blank, but not both. Use the format HH:MM:SS:FF or HH:MM:SS;FF, where HH is the hour, MM is the minute, SS is the second, and FF is the frame number. When choosing this value, take into account your setting for Input timecode source. For example, if you have embedded timecodes that start at 01:00:00:00 and you want your clip to begin five minutes into the video, use 01:05:00:00.

**Type:** string

**Required:** False

**Format:** timecode

**Pattern:** ^([^01][0-9]|2[0-4]):[0-5][0-9]:[0-5][0-9]:[0-9]{2}$

**endTimecode**

Set End timecode (EndTimecode) to the end of the portion of the input you are clipping. The frame corresponding to the End timecode value is included in the clip. Start timecode or End timecode may be left blank, but not both. Use the format HH:MM:SS:FF or HH:MM:SS;FF, where HH is the hour, MM is the minute, SS is the second, and FF is the frame number. When choosing this value, take into account your setting for timecode source under input settings (InputTimecodeSource). For example, if you have embedded timecodes that start at 01:00:00:00 and you want your clip to end six minutes into the video, use 01:06:00:00.

**Type:** string

**Required:** False

**Format:** timecode

**Pattern:** ^([^01][0-9]|2[0-4]):[0-5][0-9]:[0-5][0-9]:[0-9]{2}$

### InputDeblockFilter (enum)

Enable Deblock (InputDeblockFilter) to produce smoother motion in the output. Default is disabled. Only manually controllable for MPEG2 and uncompressed video inputs.

- ENABLED
- DISABLED

### InputDenoiseFilter (enum)

Enable Denoise (InputDenoiseFilter) to filter noise from the input. Default is disabled. Only applicable to MPEG2, H.264, H.265, and uncompressed video inputs.
**InputFilterEnable (enum)**

Use Filter enable (InputFilterEnable) to specify how the transcoding service applies the denoise and deblock filters. You must also enable the filters separately, with Denoise (InputDenoiseFilter) and Deblock (InputDeblockFilter).  
* **Auto** - The transcoding service determines whether to apply filtering, depending on input type and quality.  
* **Disable** - The input is not filtered. This is true even if you use the API to enable them in (InputDeblockFilter) and (InputDeblockFilter).  
* **Force** - The input is filtered regardless of input type.

- AUTO
- DISABLE
- FORCE

**InputPsiControl (enum)**

Set PSI control (InputPsiControl) for transport stream inputs to specify which data the demux process to scans.  
* **Ignore PSI** - Scan all PIDs for audio and video.  
* **Use PSI** - Scan only PSI data.

- IGNORE_PSI
- USE_PSI

**InputTimecodeSource (enum)**

Timecode source under input settings (InputTimecodeSource) only affects the behavior of features that apply to a single input at a time, such as input clipping and synchronizing some captions formats. Use this setting to specify whether the service counts frames by timecodes embedded in the video (EMBEDDED) or by starting the first frame at zero (ZEROBASED). In both cases, the timecode format is HH:MM:SS:FF or HH:MM:SS;FF, where FF is the frame number. Only set this to EMBEDDED if your source video has embedded timecodes.

- EMBEDDED
- ZEROBASED
- SPECIFIEDSTART

**InsertableImage**

**duration**

Use Duration (Duration) to set the time, in milliseconds, for the image to remain on the output video.

Type: integer  
Required: False  
Minimum: -2147483648  
Maximum: 2147483647

**fadeOut**

Use Fade out (FadeOut) to set the length, in milliseconds, of the inserted image fade out. If you don’t specify a value for Fade out, the image will disappear abruptly at the end of the inserted image duration.

Type: integer
imageY

Use Top (ImageY) to set the distance, in pixels, between the inserted image and the top edge of the video frame. Required for BMP, PNG and TGA input.

Type: integer
Required: True
Minimum: -2147483648
Maximum: 2147483647

fadeIn

Use Fade in (FadeIn) to set the length, in milliseconds, of the inserted image fade in. If you don't specify a value for Fade in, the image will appear abruptly at the Start time.

Type: integer
Required: False
Minimum: -2147483648
Maximum: 2147483647

imageX

Use Left (ImageX) to set the distance, in pixels, between the inserted image and the left edge of the frame. Required for BMP, PNG and TGA input.

Type: integer
Required: True
Minimum: -2147483648
Maximum: 2147483647

width

Specify the Width (Width) of the inserted image. Use a value that is less than or equal to the video resolution width. Leave this setting blank to use the native width of the image.

Type: integer
Required: False
Minimum: -2147483648
Maximum: 2147483647

startTime

Use Start time (StartTime) to specify the video timecode when the image is inserted in the output. This must be in timecode (HH:MM:SS:FF or HH:MM:SS;FF) format.

Type: string
Required: False
Pattern: ^((((0-1)\d)\d)?([0-1]\d))(2[0-3])?(:([0-5]\d)\d)\d\d)(??)[0-5]\d)\d$
opacity

Use Opacity (Opacity) to specify how much of the underlying video shows through the inserted image. 0 is transparent and 100 is fully opaque. Default is 50.

Type: integer
Required: True
Minimum: 0
Maximum: 100

layer

Use Layer (Layer) to specify how overlapping inserted images appear. Images with higher values of layer appear on top of images with lower values of layer.

Type: integer
Required: True
Minimum: 0
Maximum: 99

height

Specify the Height (Height) of the inserted image. Use a value that is less than or equal to the video resolution height. Leave this setting blank to use the native height of the image.

Type: integer
Required: False
Minimum: -2147483648
Maximum: 2147483647

imageInserterInput

Use Image location (imageInserterInput) to specify the Amazon S3 location of the image to be inserted into the output. Use a 32 bit BMP, PNG, or TGA file that fits inside the video frame.

Type: string
Required: True
Pattern: ^s3:/

Job

settings

Type: JobSettings (p. 594)
Required: True

role

The IAM role you use for creating this job. For details about permissions, see the User Guide topic at the User Guide at http://docs.aws.amazon.com/mediaconvert/latest/ug/iam-role.html

Type: string
Required: True
outputGroupDetails

List of output group details

  Type: Array of type OutputGroupDetail (p. 623)
  Required: False

jobTemplate

The job template that the job is created from, if it is created from a job template.

  Type: string
  Required: False

timing

  Type: Timing (p. 633)
  Required: False

errorMessage

Error message of Job

  Type: string
  Required: False

errorCode

Error code for the job

  Type: integer
  Required: False
  Format: int32

createdAt

The time, in Unix epoch format in seconds, when the job got created.

  Type: string
  Required: False
  Format: date-time

id

A portion of the job's ARN, unique within your AWS Elemental MediaConvert resources

  Type: string
  Required: False

arn

An identifier for this resource that is unique within all of AWS.

  Type: string
Required: False

userMetadata
User-defined metadata that you want to associate with an MediaConvert job. You specify metadata in key/value pairs.

Type: object
  Required: False

queue
Optional. When you create a job, you can specify a queue to send it to. If you don't specify, the job will go to the default queue. For more about queues, see the User Guide topic at http://docs.aws.amazon.com/mediaconvert/latest/ug/what-is.html

Type: string
  Required: False

status

Type: string
  Required: False

JobSettings

timecodeConfig
Contains settings used to acquire and adjust timecode information from inputs.

Type: TimecodeConfig (p. 632)
  Required: False

adAvailOffset
When specified, this offset (in milliseconds) is added to the input Ad Avail PTS time.

Type: integer
  Required: False
  Minimum: -1000
  Maximum: 1000

outputGroups
(OutputGroups) contains one group of settings for each set of outputs that share a common package type. All unpackaged files (MPEG-4, MPEG-2 TS, Quicktime, MXF, and no container) are grouped in a single output group as well. Required in (OutputGroups) is a group of settings that apply to the whole group. This required object depends on the value you set for (Type) under (OutputGroups)>(OutputGroupSettings). Type, settings object pairs are as follows. * FILE_GROUP_SETTINGS, FileGroupSettings * HLS_GROUP_SETTINGS, HlsGroupSettings * DASH_ISO_GROUP_SETTINGS, DashIsoGroupSettings * MS_SMOOTH_GROUP_SETTINGS, MsSmoothGroupSettings * CMAF_GROUP_SETTINGS, CmafGroupSettings

Type: Array of type OutputGroup (p. 623)
**Required**: True

**nielsenConfiguration**

*Type*: NielsenConfiguration (p. 619)  
*Required*: False

**inputs**

Use Inputs (inputs) to define source file used in the transcode job. There can be multiple inputs added in a job. These inputs will be concatenated together to create the output.

*Type*: Array of type Input (p. 587)  
*Required*: True

**timedMetadataInsertion**

*Type*: TimedMetadataInsertion (p. 633)  
*Required*: False

**availBlanking**

Settings for ad avail blanking. Video can be blanked or overlaid with an image, and audio muted during SCTE-35 triggered ad avails.

*Type*: AvailBlanking (p. 529)  
*Required*: False

**JobStatus (enum)**

A job's status can be SUBMITTED, PROGRESSING, COMPLETE, CANCELED, or ERROR.

SUBMITTED  
PROGRESSING  
COMPLETE  
CANCELED  
ERROR  

**LanguageCode (enum)**


ENG  
SPA  
FRA  
DEU  
GER  
ZHO  
ARA  
HIN  
JPN  
RUS
Properties

FIJ
FIN
FRM
FUL
GLA
GLG
LUG
KAT
ELL
GRN
GUJ
HAT
HAU
HEB
HER
HMO
HUN
ISL
IDO
IBO
IND
INA
ILE
IKU
IPK
GLE
JAV
KAL
KAN
KAU
KAS
KAZ
KIK
KIN
KIR
KOM
KON
KUA
KUR
LAO
LAT
LAV
LIM
LIN
LIT
LUB
LTZ
MKD
MLG
MSA
MAL
MLT
GLV
MRI
MAR
MAH
MON
NAU
NAV
NDE
NBL
NDO
NEP
SME
NOR
NOB
NNO
OCI
OJI
ORI
ORM
OSS
PLI
FAS
POL
PUS
QUE
QAA
RON
ROH
RUN
SMO
SAG
SAN
SRD
SRB
SNA
III
SND
SIN
SLK
SLV
SOM
SOT
SUN
SWA
SSW
SWE
TGL
TAH
TGK
TAM
TAT
TEL
M2tsAudioBufferModel (enum)

Selects between the DVB and ATSC buffer models for Dolby Digital audio.

DVB
ATSC

M2tsBufferModel (enum)

Controls what buffer model to use for accurate interleaving. If set to MULTIPLEX, use multiplex buffer model. If set to NONE, this can lead to lower latency, but low-memory devices may not be able to play back the stream without interruptions.

MULTIPLEX
NONE

M2tsEbpAudioInterval (enum)

When set to VIDEO_AND_FIXED_INTERVALS, audio EBP markers will be added to partitions 3 and 4. The interval between these additional markers will be fixed, and will be slightly shorter than the video EBP marker interval. When set to VIDEO_INTERVAL, these additional markers will not be inserted. Only applicable when EBP segmentation markers are is selected (segmentationMarkers is EBP or EBP_LEGACY).

VIDEO_AND_FIXED_INTERVALS
VIDEO_INTERVAL
M2tsEbpPlacement (enum)

Selects which PIDs to place EBP markers on. They can either be placed only on the video PID, or on both the video PID and all audio PIDs. Only applicable when EBP segmentation markers are selected (segmentationMarkers is EBP or EBP_LEGACY).

- VIDEO_AND_AUDIO_PIDS
- VIDEO_PID

M2tsEsRateInPes (enum)

Controls whether to include the ES Rate field in the PES header.

- INCLUDE
- EXCLUDE

M2tsNielsenId3 (enum)

If INSERT, Nielsen inaudible tones for media tracking will be detected in the input audio and an equivalent ID3 tag will be inserted in the output.

- INSERT
- NONE

M2tsPcrControl (enum)

When set to PCR_EVERY_PES_PACKET, a Program Clock Reference value is inserted for every Packetized Elementary Stream (PES) header. This is effective only when the PCR PID is the same as the video or audio elementary stream.

- PCR_EVERY_PES_PACKET
- CONFIGURED_PCR_PERIOD

M2tsRateMode (enum)

When set to CBR, inserts null packets into transport stream to fill specified bitrate. When set to VBR, the bitrate setting acts as the maximum bitrate, but the output will not be padded up to that bitrate.

- VBR
- CBR

M2tsScte35Source (enum)

Enables SCTE-35 passthrough (scte35Source) to pass any SCTE-35 signals from input to output.

- PASSTHROUGH
- NONE

M2tsSegmentationMarkers (enum)

Inserts segmentation markers at each segmentation_time period. rai_segstart sets the Random Access Indicator bit in the adaptation field. rai_adapt sets the RAI bit and adds the current timecode in the private data bytes. psi_segstart inserts PAT and PMT tables at the start of segments. ebp adds Encoder Boundary Point information to the adaptation field as per OpenCable specification OC-SP-EBP-
I01-130118. ebp_legacy adds Encoder Boundary Point information to the adaptation field using a legacy proprietary format.

NONE
RAISEGSTART
RAI_ADAPT
PSI_SEGSTART
EBP
EBP_LEGACY

**M2tsSegmentationStyle (enum)**

The segmentation style parameter controls how segmentation markers are inserted into the transport stream. With avails, it is possible that segments may be truncated, which can influence where future segmentation markers are inserted. When a segmentation style of "reset_cadence" is selected and a segment is truncated due to an avail, we will reset the segmentation cadence. This means the subsequent segment will have a duration of $segmentation_time seconds. When a segmentation style of "maintain_cadence" is selected and a segment is truncated due to an avail, we will not reset the segmentation cadence. This means the subsequent segment will likely be truncated as well. However, all segments after that will have a duration of $segmentation_time seconds. Note that EBP lookahead is a slight exception to this rule.

MAINTAIN_CADENCE
RESET_CADENCE

**M2tsSettings**

dvbTeletextPid

Packet Identifier (PID) for input source DVB Teletext data to this output.

**Type:** integer  
**Required:** False  
**Minimum:** 32  
**Maximum:** 8182

nielsenId3

**Type:** string  
**Required:** False

bitrate

The output bitrate of the transport stream in bits per second. Setting to 0 lets the muxer automatically determine the appropriate bitrate. Other common values are 3750000, 7500000, and 15000000.

**Type:** integer  
**Required:** False  
**Minimum:** 0  
**Maximum:** 2147483647

segmentationTime

The length in seconds of each segment. Required unless markers is set to _none_.

---

601
Properties

**Type**: number
**Required**: False
**Format**: float
**Minimum**: 0.0

**audioPids**
Packet Identifier (PID) of the elementary audio stream(s) in the transport stream. Multiple values are accepted, and can be entered in ranges and/or by comma separation.

**Type**: Array of type integer
**Required**: False

**rateMode**

**Type**: string
**Required**: False

**maxPcrInterval**
Maximum time in milliseconds between Program Clock References (PCRs) inserted into the transport stream.

**Type**: integer
**Required**: False
**Minimum**: 0
**Maximum**: 500

**audioFramesPerPes**
The number of audio frames to insert for each PES packet.

**Type**: integer
**Required**: False
**Minimum**: 0
**Maximum**: 2147483647

**ebpAudioInterval**

**Type**: string
**Required**: False

**fragmentTime**
The length in seconds of each fragment. Only used with EBP markers.

**Type**: number
**Required**: False
**Format**: float
**Minimum**: 0.0

**scte35Pid**
Packet Identifier (PID) of the SCTE-35 stream in the transport stream.
Properties

Type: integer
Required: False
Minimum: 32
Maximum: 8182

pmtInterval

The number of milliseconds between instances of this table in the output transport stream.

Type: integer
Required: False
Minimum: 0
Maximum: 1000

privateMetadataPid

Packet Identifier (PID) of the private metadata stream in the transport stream.

Type: integer
Required: False
Minimum: 32
Maximum: 8182

segmentationStyle

Type: string
Required: False

audioBufferModel

Type: string
Required: False

programNumber

The value of the program number field in the Program Map Table.

Type: integer
Required: False
Minimum: 0
Maximum: 65535

dvbNitSettings

Type: DvbNitSettings (p. 546)
Required: False

timedMetadataPid

Packet Identifier (PID) of the timed metadata stream in the transport stream.

Type: integer
**Properties**

**Required**: False  
**Minimum**: 32  
**Maximum**: 8182

**scte35Source**

*Type*: string  
*Required*: False

**pmtPid**

Packet Identifier (PID) for the Program Map Table (PMT) in the transport stream.

*Type*: integer  
*Required*: False  
**Minimum**: 32  
**Maximum**: 8182

**bufferModel**

*Type*: string  
*Required*: False

**ebpPlacement**

*Type*: string  
*Required*: False

**dvbSdtSettings**

*Type*: DvbSdtSettings (p. 546)  
*Required*: False

**nullPacketBitrate**

Value in bits per second of extra null packets to insert into the transport stream. This can be used if a downstream encryption system requires periodic null packets.

*Type*: number  
*Required*: False  
*Format*: float  
**Minimum**: 0.0

**pcrPid**

Packet Identifier (PID) of the Program Clock Reference (PCR) in the transport stream. When no value is given, the encoder will assign the same value as the Video PID.

*Type*: integer  
*Required*: False  
**Minimum**: 32  
**Maximum**: 8182
**minEbpInterval**

When set, enforces that Encoder Boundary Points do not come within the specified time interval of each other by looking ahead at input video. If another EBP is going to come in within the specified time interval, the current EBP is not emitted, and the segment is "stretched" to the next marker. The lookahead value does not add latency to the system. The Live Event must be configured elsewhere to create sufficient latency to make the lookahead accurate.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 10000

**transportStreamId**

The value of the transport stream ID field in the Program Map Table.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 65535

**videoPid**

Packet Identifier (PID) of the elementary video stream in the transport stream.

- **Type**: integer
- **Required**: False
- **Minimum**: 32
- **Maximum**: 8182

**pcrControl**

- **Type**: string
- **Required**: False

**esRateInPes**

- **Type**: string
- **Required**: False

**segmentationMarkers**

- **Type**: string
- **Required**: False

**dvbTdtSettings**

- **Type**: [DvbTdtSettings](p. 551)
- **Required**: False

**patInterval**

The number of milliseconds between instances of this table in the output transport stream.
**Type**: integer  
**Required**: False  
**Minimum**: 0  
**Maximum**: 1000

**dvbSubPids**

Packet Identifier (PID) for input source DVB Subtitle data to this output. Multiple values are accepted, and can be entered in ranges and/or by comma separation.

**Type**: Array of type integer  
**Required**: False

**M3u8NielsenId3 (enum)**

If INSERT, Nielsen inaudible tones for media tracking will be detected in the input audio and an equivalent ID3 tag will be inserted in the output.

- INSERT
- NONE

**M3u8PcrControl (enum)**

When set to PCR_EVERY_PES_PACKET a Program Clock Reference value is inserted for every Packetized Elementary Stream (PES) header. This parameter is effective only when the PCR PID is the same as the video or audio elementary stream.

- PCR_EVERY_PES_PACKET
- CONFIGURED_PCR_PERIOD

**M3u8Scte35Source (enum)**

Enables SCTE-35 passthrough (scte35Source) to pass any SCTE-35 signals from input to output.

- PASSTHROUGH
- NONE

**M3u8Settings**

**pmtPid**

Packet Identifier (PID) for the Program Map Table (PMT) in the transport stream.

**Type**: integer  
**Required**: False  
**Minimum**: 32  
**Maximum**: 8182

**nielsenId3**

**Type**: string  
**Required**: False
Properties

**pcrPid**
Packet Identifier (PID) of the Program Clock Reference (PCR) in the transport stream. When no value is given, the encoder will assign the same value as the Video PID.

- **Type**: integer
- **Required**: False
- **Minimum**: 32
- **Maximum**: 8182

**audioPids**
Packet Identifier (PID) of the elementary audio stream(s) in the transport stream. Multiple values are accepted, and can be entered in ranges and/or by comma separation.

- **Type**: Array of type integer
- **Required**: False

**audioFramesPerPes**
The number of audio frames to insert for each PES packet.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 2147483647

**scte35Pid**
Packet Identifier (PID) of the SCTE-35 stream in the transport stream.

- **Type**: integer
- **Required**: False
- **Minimum**: 32
- **Maximum**: 8182

**transportStreamId**
The value of the transport stream ID field in the Program Map Table.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 65535

**pcrControl**

- **Type**: string
- **Required**: False

**videoPid**
Packet Identifier (PID) of the elementary video stream in the transport stream.
**Properties**

Type: integer  
Required: False  
Minimum: 32  
Maximum: 8182

**privateMetadataPid**

Packet Identifier (PID) of the private metadata stream in the transport stream.

Type: integer  
Required: False  
Minimum: 32  
Maximum: 8182

**pmtInterval**

The number of milliseconds between instances of this table in the output transport stream.

Type: integer  
Required: False  
Minimum: 0  
Maximum: 1000

**patInterval**

The number of milliseconds between instances of this table in the output transport stream.

Type: integer  
Required: False  
Minimum: 0  
Maximum: 1000

**programNumber**

The value of the program number field in the Program Map Table.

Type: integer  
Required: False  
Minimum: 0  
Maximum: 65535

**timedMetadataPid**

Packet Identifier (PID) of the timed metadata stream in the transport stream.

Type: integer  
Required: False  
Minimum: 32  
Maximum: 8182

**timedMetadata**

Type: string
**Properties**

**Required**: False

**scte35Source**

*Type*: string  
*Required*: False

**MovClapAtom (enum)**

When enabled, include 'clap' atom if appropriate for the video output settings.

- **INCLUDE**
- **EXCLUDE**

**MovCslgAtom (enum)**

When enabled, file composition times will start at zero, composition times in the 'ctts' (composition time to sample) box for B-frames will be negative, and a 'cslg' (composition shift least greatest) box will be included per 14496-1 amendment 1. This improves compatibility with Apple players and tools.

- **INCLUDE**
- **EXCLUDE**

**MovMpeg2FourCCControl (enum)**

When set to XDCAM, writes MPEG2 video streams into the QuickTime file using XDCAM fourcc codes. This increases compatibility with Apple editors and players, but may decrease compatibility with other players. Only applicable when the video codec is MPEG2.

- **XDCAM**
- **MPEG**

**MovPaddingControl (enum)**

If set to OMNEON, inserts Omneon-compatible padding

- **OMNEON**
- **NONE**

**MovReference (enum)**

A value of 'external' creates separate media files and the wrapper file (.mov) contains references to these media files. A value of 'self_contained' creates only a wrapper (.mov) file and this file contains all of the media.

- **SELF_CONTAINED**
- **EXTERNAL**

**MovSettings**

*reference*

*Type*: string
paddingControl

  Type: string
  Required: False

mpeg2FourCCControl

  Type: string
  Required: False

cslgAtom

  Type: string
  Required: False

clapAtom

  Type: string
  Required: False

Mp2Settings

channels

  Set Channels to specify the number of channels in this output audio track. Choosing Mono in the console
  will give you 1 output channel; choosing Stereo will give you 2. In the API, valid values are 1 and 2.

    Type: integer
    Required: False
    Minimum: 1
    Maximum: 2

bitrate

  Average bitrate in bits/second.

    Type: integer
    Required: False
    Minimum: 32000
    Maximum: 384000

sampleRate

  Sample rate in hz.

    Type: integer
    Required: False
    Minimum: 32000
    Maximum: 48000
**Mp4CslgAtom (enum)**

When enabled, file composition times will start at zero, composition times in the 'ctts' (composition time to sample) box for B-frames will be negative, and a 'cslg' (composition shift least greatest) box will be included per 14496-1 amendment 1. This improves compatibility with Apple players and tools.

- INCLUDE
- EXCLUDE

**Mp4FreeSpaceBox (enum)**

Inserts a free-space box immediately after the moov box.

- INCLUDE
- EXCLUDE

**Mp4MoovPlacement (enum)**

If set to PROGRESSIVE_DOWNLOAD, the MOOV atom is relocated to the beginning of the archive as required for progressive downloading. Otherwise it is placed normally at the end.

- PROGRESSIVE_DOWNLOAD
- NORMAL

**Mp4Settings**

**mp4MajorBrand**

Overrides the "Major Brand" field in the output file. Usually not necessary to specify.

- Type: string
- Required: False

**moovPlacement**

- Type: string
- Required: False

**cslgAtom**

- Type: string
- Required: False

**freeSpaceBox**

- Type: string
- Required: False

**Mpeg2AdaptiveQuantization (enum)**

Adaptive quantization. Allows intra-frame quantizers to vary to improve visual quality.

- OFF
Mpeg2CodecLevel (enum)
Use Level (Mpeg2CodecLevel) to set the MPEG-2 level for the video output.

AUTO
LOW
MAIN
HIGH
HIGH1440

Mpeg2CodecProfile (enum)
Use Profile (Mpeg2CodecProfile) to set the MPEG-2 profile for the video output.

MAIN
PROFILE_422

Mpeg2FramerateControl (enum)
If you are using the console, use the Framerate setting to specify the framerate for this output. If you want to keep the same framerate as the input video, choose Follow source. If you want to do framerate conversion, choose a framerate from the dropdown list or choose Custom. The framerates shown in the dropdown list are decimal approximations of fractions. If you choose Custom, specify your framerate as a fraction. If you are creating your transcoding job specification as a JSON file without the console, use FramerateControl to specify which value the service uses for the framerate for this output. Choose INITIALIZE_FROM_SOURCE if you want the service to use the framerate from the input. Choose SPECIFIED if you want the service to use the framerate you specify in the settings FramerateNumerator and FramerateDenominator.

INITIALIZE_FROM_SOURCE
SPECIFIED

Mpeg2FramerateConversionAlgorithm (enum)
When set to INTERPOLATE, produces smoother motion during framerate conversion.

DUPLICATE_DROP
INTERPOLATE

Mpeg2GopSizeUnits (enum)
Indicates if the GOP Size in MPEG2 is specified in frames or seconds. If seconds the system will convert the GOP Size into a frame count at run time.

FRAMES
SECONDS

Mpeg2InterlaceMode (enum)
Use Interlace mode (InterlaceMode) to choose the scan line type for the output. * Top Field First (TOP_FIELD) and Bottom Field First (BOTTOM_FIELD) produce interlaced output with the entire output
having the same field polarity (top or bottom first). * Follow, Default Top (FOLLOW_TOP_FIELD) and Follow, Default Bottom (FOLLOW_BOTTOM_FIELD) use the same field polarity as the source. Therefore, behavior depends on the input scan type. - If the source is interlaced, the output will be interlaced with the same polarity as the source (it will follow the source). The output could therefore be a mix of “top field first” and “bottom field first”. - If the source is progressive, the output will be interlaced with “top field first” or “bottom field first” polarity, depending on which of the Follow options you chose.

PROGRESSIVE
TOP_FIELD
BOTTOM_FIELD
FOLLOW_TOP_FIELD
FOLLOW_BOTTOM_FIELD

Mpeg2IntraDcPrecision (enum)

Use Intra DC precision (Mpeg2IntraDcPrecision) to set quantization precision for intra-block DC coefficients. If you choose the value auto, the service will automatically select the precision based on the per-frame compression ratio.

AUTO
INTRA_DC_PRECISION_8
INTRA_DC_PRECISION_9
INTRA_DC_PRECISION_10
INTRA_DC_PRECISION_11

Mpeg2ParControl (enum)

Using the API, enable ParFollowSource if you want the service to use the pixel aspect ratio from the input. Using the console, do this by choosing Follow source for Pixel aspect ratio.

INITIALIZE_FROM_SOURCE
SPECIFIED

Mpeg2QualityTuningLevel (enum)

Use Quality tuning level (Mpeg2QualityTuningLevel) to specify whether to use single-pass or multipass video encoding.

SINGLE_PASS
MULTI_PASS

Mpeg2RateControlMode (enum)

Use Rate control mode (Mpeg2RateControlMode) to specify whether the bitrate is variable (vbr) or constant (cbr).

VBR
CBR

Mpeg2SceneChangeDetect (enum)

Scene change detection (inserts I-frames on scene changes).

DISABLED
ENABLED

**Mpeg2Settings**

**minIInterval**

Enforces separation between repeated (cadence) I-frames and I-frames inserted by Scene Change Detection. If a scene change I-frame is within I-interval frames of a cadence I-frame, the GOP is shrunk and/or stretched to the scene change I-frame. GOP stretch requires enabling lookahead as well as setting I-interval. The normal cadence resumes for the next GOP. This setting is only used when Scene Change Detect is enabled. Note: Maximum GOP stretch = GOP size + Min-I-interval - 1

*Type:* integer  
*Required:* False  
*Minimum:* 0  
*Maximum:* 30

**parNumerator**

Pixel Aspect Ratio numerator.

*Type:* integer  
*Required:* False  
*Minimum:* 1  
*Maximum:* 2147483647

**gopSizeUnits**

*Type:* string  
*Required:* False

**hrdBUFFERSize**

Size of buffer (HRD buffer model) in bits. For example, enter five megabits as 5000000.

*Type:* integer  
*Required:* False  
*Minimum:* 0  
*Maximum:* 47185920

**qualityTuningLevel**

*Type:* string  
*Required:* False

**maxBitrate**

Maximum bitrate in bits/second. For example, enter five megabits per second as 5000000.

*Type:* integer  
*Required:* False  
*Minimum:* 1000  
*Maximum:* 300000000
Properties

**bitrate**

Average bitrate in bits/second. Required for VBR and CBR. For MS Smooth outputs, bitrates must be unique when rounded down to the nearest multiple of 1000.

- **Type**: integer
- **Required**: False
- **Minimum**: 1000
- **Maximum**: 288000000

**spatialAdaptiveQuantization**

- **Type**: string
- **Required**: False

**slowPal**

- **Type**: string
- **Required**: False

**intraDcPrecision**

- **Type**: string
- **Required**: False

**codecProfile**

- **Type**: string
- **Required**: False

**softness**

Softness. Selects quantizer matrix, larger values reduce high-frequency content in the encoded image.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 128

**framerateControl**

- **Type**: string
- **Required**: False

**telecine**

- **Type**: string
- **Required**: False

**codecLevel**

- **Type**: string
Properties

Required: False

framerateConversionAlgorithm

Type: string
Required: False

temporalAdaptiveQuantization

Type: string
Required: False

hrdBufferInitialFillPercentage

Percentage of the buffer that should initially be filled (HRD buffer model).

Type: integer
Required: False
Minimum: 0
Maximum: 100

framerateNumerator

Framerate numerator - framerate is a fraction, e.g. 24000 / 1001 = 23.976 fps.

Type: integer
Required: False
Minimum: 24
Maximum: 60000

gopClosedCadence

Frequency of closed GOPs. In streaming applications, it is recommended that this be set to 1 so a decoder joining mid-stream will receive an IDR frame as quickly as possible. Setting this value to 0 will break output segmenting.

Type: integer
Required: False
Minimum: 0
Maximum: 2147483647

numberBFramesBetweenReferenceFrames

Number of B-frames between reference frames.

Type: integer
Required: False
Minimum: 0
Maximum: 7

framerateDenominator

Framerate denominator.
- **Type**: integer
  - **Required**: False
  - **Minimum**: 1
  - **Maximum**: 1001

- **adaptiveQuantization**
  - **Type**: string
  - **Required**: False

- **interlaceMode**
  - **Type**: string
  - **Required**: False

- **gopSize**
  - **Type**: number
  - **Required**: False
  - **Format**: float
  - **Minimum**: 0.0

- **syntax**
  - **Type**: string
  - **Required**: False

- **parDenominator**
  - **Type**: integer
  - **Required**: False
  - **Minimum**: 1
  - **Maximum**: 2147483647

- **sceneChangeDetect**
  - **Type**: string
  - **Required**: False

- **parControl**
  - **Type**: string
  - **Required**: False

- **rateControlMode**
  - **Type**: string
  - **Required**: False
**Mpeg2SlowPal (enum)**

Enables Slow PAL rate conversion. 23.976fps and 24fps input is relabeled as 25fps, and audio is sped up correspondingly.

- DISABLED
- ENABLED

**Mpeg2SpatialAdaptiveQuantization (enum)**

Adjust quantization within each frame based on spatial variation of content complexity.

- DISABLED
- ENABLED

**Mpeg2Syntax (enum)**

Produces a Type D-10 compatible bitstream (SMPTE 356M-2001).

- DEFAULT
- D_10

**Mpeg2Telecine (enum)**

Only use Telecine (Mpeg2Telecine) when you set Framerate (Framerate) to 29.970. Set Telecine (Mpeg2Telecine) to Hard (hard) to produce a 29.97i output from a 23.976 input. Set it to Soft (soft) to produce 23.976 output and leave conversion to the player.

- NONE
- SOFT
- HARD

**Mpeg2TemporalAdaptiveQuantization (enum)**

Adjust quantization within each frame based on temporal variation of content complexity.

- DISABLED
- ENABLED

**MsSmoothAudioDeduplication (enum)**

COMBINE_DUPLICATE_STREAMS combines identical audio encoding settings across a Microsoft Smooth output group into a single audio stream.

- COMBINE_DUPLICATE_STREAMS
- NONE

**MsSmoothEncryptionSettings**

**spekeKeyProvider**

- **Type**: SpekeKeyProvider (p. 630)
- **Required**: True
MsSmoothGroupSettings

fragmentLength

Use Fragment length (FragmentLength) to specify the mp4 fragment sizes in seconds. Fragment length must be compatible with GOP size and framerate.

Type: integer
Required: True
Minimum: 1
Maximum: 2147483647

encryption

Type: MsSmoothEncryptionSettings (p. 618)
Required: False

audioDeduplication

Type: string
Required: False

manifestEncoding

Type: string
Required: False

destination

Use Destination (Destination) to specify the S3 output location and the output filename base. Destination accepts format identifiers. If you do not specify the base filename in the URI, the service will use the filename of the input file. If your job has multiple inputs, the service uses the filename of the first input file.

Type: string
Required: False
Pattern: ^s3:/\/

MsSmoothManifestEncoding (enum)

Use Manifest encoding (MsSmoothManifestEncoding) to specify the encoding format for the server and client manifest. Valid options are utf8 and utf16.

UTF8
UTF16

NielsenConfiguration

distributorId

Use Distributor ID (DistributorID) to specify the distributor ID that is assigned to your organization by Nielsen.

Type: string
Required: False

**breakoutCode**

Use Nielsen Configuration (NielsenConfiguration) to set the Nielsen measurement system breakout code. Supported values are 0, 3, 7, and 9.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 9

**NoiseReducer**

**filter**

- **Type**: string
- **Required**: True

**filterSettings**

- **Type**: NoiseReducerFilterSettings (p. 620)
- **Required**: False

**spatialFilterSettings**

- **Type**: NoiseReducerSpatialFilterSettings (p. 621)
- **Required**: False

**NoiseReducerFilter (enum)**

Use Noise reducer filter (NoiseReducerFilter) to select one of the following spatial image filtering functions. To use this setting, you must also enable Noise reducer (NoiseReducer). *Bilateral* is an edge preserving noise reduction filter. *Mean* (softest), Gaussian, Lanczos, and *Sharpen* (sharpest) are convolution filters. *Conserve* is a min/max noise reduction filter. *Spatial* is a frequency-domain filter based on JND principles.

- **BILATERAL**
- **MEAN**
- **GAUSSIAN**
- **LANCZOS**
- **SHARPEN**
- **CONSERVE**
- **SPATIAL**

**NoiseReducerFilterSettings**

**strength**

Relative strength of noise reducing filter. Higher values produce stronger filtering.

- **Type**: integer
- **Required**: False
**Minimum**: 0  
**Maximum**: 3

**NoiseReducerSpatialFilterSettings**

**strength**
Relative strength of noise reducing filter. Higher values produce stronger filtering.

Type: integer  
Required: False  
Minimum: 0  
Maximum: 16

**postFilterSharpenStrength**
Specify strength of post noise reduction sharpening filter, with 0 disabling the filter and 3 enabling it at maximum strength.

Type: integer  
Required: False  
Minimum: 0  
Maximum: 3

**speed**
The speed of the filter, from -2 (lower speed) to 3 (higher speed), with 0 being the nominal value.

Type: integer  
Required: False  
Minimum: -2  
Maximum: 3

**Output**

**extension**
Use Extension (Extension) to specify the file extension for outputs in File output groups. If you do not specify a value, the service will use default extensions by container type as follows: * MPEG-2 transport stream, m2ts  * Quicktime, mov  * MXF container, mxf  * MPEG-4 container, mp4  * No Container, the service will use codec extensions (e.g. AAC, H265, H265, AC3)

Type: string  
Required: False

**videoDescription**
(VideoDescription) contains a group of video encoding settings. The specific video settings depend on the video codec you choose when you specify a value for Video codec (codec). Include one instance of (VideoDescription) per output.

Type: VideoDescription (p. 635)  
Required: False
audioDescriptions

(AudioDescriptions) contains groups of audio encoding settings organized by audio codec. Include one instance of (AudioDescriptions) per output. (AudioDescriptions) can contain multiple groups of encoding settings.

- **Type**: Array of type [AudioDescription](p. 526)
- **Required**: False

containerSettings

- **Type**: [ContainerSettings](p. 542)
- **Required**: False

preset

Use Preset (Preset) to specify a preset for your transcoding settings. Provide the system or custom preset name. You can specify either Preset (Preset) or Container settings (ContainerSettings), but not both.

- **Type**: string
- **Required**: False

outputSettings

- **Type**: [OutputSettings](p. 625)
- **Required**: False

captionDescriptions

(CaptionDescriptions) contains groups of captions settings. For each output that has captions, include one instance of (CaptionDescriptions). (CaptionDescriptions) can contain multiple groups of captions settings.

- **Type**: Array of type [CaptionDescription](p. 533)
- **Required**: False

nameModifier

Use Name modifier (NameModifier) to have the service add a string to the end of each output filename. You specify the base filename as part of your destination URI. When you create multiple outputs in the same output group, Name modifier (NameModifier) is required. Name modifier also accepts format identifiers. For DASH ISO outputs, if you use the format identifiers $Number$ or $Time$ in one output, you must use them in the same way in all outputs of the output group.

- **Type**: string
- **Required**: False

OutputChannelMapping

inputChannels

List of input channels

- **Type**: Array of type integer
OutputDetail

durationInMs
Duration in milliseconds

  Type: integer
  Required: False

videoDetails

  Type: VideoDetail (p. 637)
  Required: False

OutputGroup

outputs
This object holds groups of encoding settings, one group of settings per output.

  Type: Array of type Output (p. 621)
  Required: True

outputGroupSettings

  Type: OutputGroupSettings (p. 624)
  Required: True

name
Name of the output group

  Type: string
  Required: False

customName
Use Custom Group Name (CustomName) to specify a name for the output group. This value is displayed on the console and can make your job settings JSON more human-readable. It does not affect your outputs. Use up to twelve characters that are either letters, numbers, spaces, or underscores.

  Type: string
  Required: False

OutputGroupDetail

outputDetails
Details about the output
Type: Array of type `OutputDetail (p. 623)`
Required: False

**OutputGroupSettings**

**dashIsoGroupSettings**

Type: `DashIsoGroupSettings (p. 543)`
Required: False

**fileGroupSettings**

Type: `FileGroupSettings (p. 557)`
Required: False

**msSmoothGroupSettings**

Type: `MsSmoothGroupSettings (p. 619)`
Required: False

**cmafGroupSettings**

Type: `CmafGroupSettings (p. 537)`
Required: False

**type**

Type: string
Required: True

**hlsGroupSettings**

Type: `HlsGroupSettings (p. 581)`
Required: False

**OutputGroupType (enum)**

Type of output group (File group, Apple HLS, DASH ISO, Microsoft Smooth Streaming, CMAF)

- `HLS_GROUP_SETTINGS`
- `DASH_ISO_GROUP_SETTINGS`
- `FILE_GROUP_SETTINGS`
- `MS_SMOOTH_GROUP_SETTINGS`
- `CMAF_GROUP_SETTINGS`

**OutputSdt (enum)**

Selects method of inserting SDT information into output stream. "Follow input SDT" copies SDT information from input stream to output stream. "Follow input SDT if present" copies SDT information from input stream to output stream if SDT information is present in the input, otherwise it will fall back on the user-defined values. Enter "SDT Manually" means user will enter the SDT information. "No SDT" means output stream will not contain SDT information.
**OutputSettings**

**hlsSettings**

*Type:* HlsSettings (p. 585)  
*Required:* False

**ProresCodecProfile (enum)**

Use Profile (ProResCodecProfile) to specify the type of Apple ProRes codec to use for this output.

APPLE_PRORES_422  
APPLE_PRORES_422_HQ  
APPLE_PRORES_422_LT  
APPLE_PRORES_422_PROXY

**ProresFramerateControl (enum)**

If you are using the console, use the Framerate setting to specify the framerate for this output. If you want to keep the same framerate as the input video, choose Follow source. If you want to do framerate conversion, choose a framerate from the dropdown list or choose Custom. The framerates shown in the dropdown list are decimal approximations of fractions. If you choose Custom, specify your framerate as a fraction. If you are creating your transcoding job specification as a JSON file without the console, use FramerateControl to specify which value the service uses for the framerate for this output. Choose INITIALIZE_FROM_SOURCE if you want the service to use the framerate from the input. Choose SPECIFIED if you want the service to use the framerate you specify in the settings FramerateNumerator and FramerateDenominator.

INITIALIZE_FROM_SOURCE  
SPECIFIED

**ProresFramerateConversionAlgorithm (enum)**

When set to INTERPOLATE, produces smoother motion during framerate conversion.

DUPLICATE_DROP  
INTERPOLATE

**ProresInterlaceMode (enum)**

Use Interlace mode (InterlaceMode) to choose the scan line type for the output. * Top Field First (TOP_FIELD) and Bottom Field First (BOTTOM_FIELD) produce interlaced output with the entire output having the same field polarity (top or bottom first). * Follow, Default Top (FOLLOW_TOP_FIELD) and Follow, Default Bottom (FOLLOW_BOTTOM_FIELD) use the same field polarity as the source. Therefore, behavior depends on the input scan type. - If the source is interlaced, the output will be interlaced with the same polarity as the source (it will follow the source). The output could therefore be a mix of "top field first" and "bottom field first". - If the source is progressive, the output will be interlaced with "top field first" or "bottom field first" polarity, depending on which of the Follow options you chose.
PROGRESSIVE
TOP_FIELD
BOTTOM_FIELD
FOLLOW_TOP_FIELD
FOLLOW_BOTTOM_FIELD

ProresParControl (enum)

Use (ProresParControl) to specify how the service determines the pixel aspect ratio. Set to Follow source (INITIALIZE_FROM_SOURCE) to use the pixel aspect ratio from the input. To specify a different pixel aspect ratio: Using the console, choose it from the dropdown menu. Using the API, set ProresParControl to (SPECIFIED) and provide for (ParNumerator) and (ParDenominator).

    INITIALIZE_FROM_SOURCE
    SPECIFIED

ProresSettings

framerateDenominator

Framerate denominator.

    Type: integer
    Required: False
    Minimum: 1
    Maximum: 2147483647

slowPal

    Type: string
    Required: False

framerateControl

    Type: string
    Required: False

telecine

    Type: string
    Required: False

framerateConversionAlgorithm

    Type: string
    Required: False

interlaceMode

    Type: string
    Required: False
parNumerator

Pixel Aspect Ratio numerator.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 2147483647

codecProfile

- **Type:** string
- **Required:** False

parDenominator

Pixel Aspect Ratio denominator.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 2147483647

parControl

- **Type:** string
- **Required:** False

framerateNumerator

When you use the API for transcode jobs that use framerate conversion, specify the framerate as a fraction. For example, 24000 / 1001 = 23.976 fps. Use FramerateNumerator to specify the numerator of this fraction. In this example, use 24000 for the value of FramerateNumerator.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 2147483647

ProresSlowPal (enum)

Enables Slow PAL rate conversion. 23.976fps and 24fps input is relabeled as 25fps, and audio is sped up correspondingly.

- **DISABLED**
- **ENABLED**

ProresTelecine (enum)

Only use Telecine (ProresTelecine) when you set Framerate (Framerate) to 29.970. Set Telecine (ProresTelecine) to Hard (hard) to produce a 29.97i output from a 23.976 input. Set it to Soft (soft) to produce 23.976 output and leave conversion to the player.

- **NONE**
HARD

**Rectangle**

**width**
Width of rectangle in pixels.

- **Type**: integer
- **Required**: True
- **Minimum**: -2147483648
- **Maximum**: 2147483647

**x**
The distance, in pixels, between the rectangle and the left edge of the video frame.

- **Type**: integer
- **Required**: True
- **Minimum**: -2147483648
- **Maximum**: 2147483647

**y**
The distance, in pixels, between the rectangle and the top edge of the video frame.

- **Type**: integer
- **Required**: True
- **Minimum**: -2147483648
- **Maximum**: 2147483647

**height**
Height of rectangle in pixels.

- **Type**: integer
- **Required**: True
- **Minimum**: -2147483648
- **Maximum**: 2147483647

**RemixSettings**

**channelsOut**
Specify the number of channels in this output after remixing. Valid values: 1, 2, 4, 6, 8

- **Type**: integer
- **Required**: True
- **Minimum**: 1
- **Maximum**: 8

**channelMapping**

- **Type**: `ChannelMapping` (p. 536)
**Required**: True

**channelsIn**

Specify the number of audio channels from your input that you want to use in your output. With remixing, you might combine or split the data in these channels, so the number of channels in your final output might be different.

- **Type**: integer
- **Required**: True
- **Minimum**: 1
- **Maximum**: 16

**RespondToAfd (enum)**

Use Respond to AFD (RespondToAfd) to specify how the service changes the video itself in response to AFD values in the input. * Choose Respond to clip the input video frame according to the AFD value, input display aspect ratio, and output display aspect ratio. * Choose Passthrough to include the input AFD values. Do not choose this when AfdSignaling is set to (NONE). A preferred implementation of this workflow is to set RespondToAfd to (NONE) and set AfdSignaling to (AUTO). * Choose None to remove all input AFD values from this output.

- NONE
- RESPOND
- PASSTHROUGH

**ScalingBehavior (enum)**

Applies only if your input aspect ratio is different from your output aspect ratio. Enable Stretch to output (StretchToOutput) to have the service stretch your video image to fit. Leave this setting disabled to allow the service to letterbox your video instead. This setting overrides any positioning value you specify elsewhere in the job.

- DEFAULT
- STRETCH_TO_OUTPUT

**SccDestinationFramerate (enum)**

Set Framerate (SccDestinationFramerate) to make sure that the captions and the video are synchronized in the output. Specify a framerate that matches the framerate of the associated video. If the video framerate is 29.97, choose 29.97 dropframe (FRAMERATE_29_97_DROPFRAME) only if the video has video_insertion=true and drop_frame_timecode=true; otherwise, choose 29.97 non-dropframe (FRAMERATE_29_97_NON_DROPFRAME).

- FRAMERATE_23_97
- FRAMERATE_24
- FRAMERATE_29_97_DROPFRAME
- FRAMERATE_29_97_NON_DROPFRAME

**SccDestinationSettings**

- **framerate**
  - **Type**: string
Properties

**SpekeKeyProvider**

**resourceld**

The SPEKE-compliant server uses Resource ID (Resourceld) to identify content.

- **Type:** string
- **Required:** True
- **Pattern:** `^[\w-]+$`

**systemIds**

Relates to SPEKE implementation. DRM system identifiers. DASH output groups support a max of two system ids. Other group types support one system id.

- **Type:** Array of type string
- **Required:** True

**url**

Use URL (Url) to specify the SPEKE-compliant server that will provide keys for content.

- **Type:** string
- **Required:** True
- **Format:** uri
- **Pattern:** `^https://\$/`

**StaticKeyProvider**

**keyFormatVersions**

Relates to DRM implementation. Either a single positive integer version value or a slash delimited list of version values (1/2/3).

- **Type:** string
- **Required:** False
- **Pattern:** `^([\d+](/\d+)* )+$`

**keyFormat**

Relates to DRM implementation. Sets the value of the KEYFORMAT attribute. Must be 'identity' or a reverse DNS string. May be omitted to indicate an implicit value of 'identity'.

- **Type:** string
- **Required:** False
- **Pattern:** `^(identity|[A-Za-z]{2,6}([.][A-Za-z0-9-]{1,63})+ )+$`

**staticKeyValue**

Relates to DRM implementation. Use a 32-character hexadecimal string to specify Key Value (StaticKeyValue).
Properties

url

Relates to DRM implementation. The location of the license server used for protecting content.

    Type: string
    Required: True
    Format: uri

TeletextDestinationSettings

pageNumber

Set pageNumber to the Teletext page number for the destination captions for this output. This value must be a three-digit hexadecimal string; strings ending in -FF are invalid. If you are passing through the entire set of Teletext data, do not use this field.

    Type: string
    Required: False
    Pattern: ^[1-8][0-9a-fA-F][0-9a-eA-E]$

TeletextSourceSettings

pageNumber

Use Page Number (PageNumber) to specify the three-digit hexadecimal page number that will be used for Teletext captions. Do not use this setting if you are passing through teletext from the input source to output.

    Type: string
    Required: False
    Pattern: ^[1-8][0-9a-fA-F][0-9a-eA-E]$

TimecodeBurnin

prefix

Use Prefix (Prefix) to place ASCII characters before any burned-in timecode. For example, a prefix of "EZ-" will result in the timecode "EZ-00:00:00:00". Provide either the characters themselves or the ASCII code equivalents. The supported range of characters is 0x20 through 0x7e. This includes letters, numbers, and all special characters represented on a standard English keyboard.

    Type: string
    Required: False
    Pattern: ^[ -~]+$  

fontSize

Use Font Size (FontSize) to set the font size of any burned-in timecode. Valid values are 10, 16, 32, 48.

    Type: integer
**Properties**

**Required**: False  
**Minimum**: 10  
**Maximum**: 48

**position**

*Type*: string  
*Required*: False

**TimecodeBurninPosition (enum)**

Use Position (Position) under under Timecode burn-in (TimecodeBurnIn) to specify the location the burned-in timecode on output video.

- TOP_CENTER
- TOP_LEFT
- TOP_RIGHT
- MIDDLE_LEFT
- MIDDLE_CENTER
- MIDDLE_RIGHT
- BOTTOM_LEFT
- BOTTOM_CENTER
- BOTTOM_RIGHT

**TimecodeConfig**

**timestampOffset**

Only applies to outputs that support program-date-time stamp. Use Timestamp offset (TimestampOffset) to overwrite the timecode date without affecting the time and frame number. Provide the new date as a string in the format "yyyy-mm-dd". To use Time stamp offset, you must also enable Insert program-date-time (InsertProgramDateTime) in the output settings. For example, if the date part of your timecodes is 2002-1-25 and you want to change it to one year later, set Timestamp offset (TimestampOffset) to 2003-1-25.

*Type*: string  
*Required*: False  
*Pattern*: `^([0-9]{4})-(0[1-9]|1[0-2])-(0[1-9]|1[2][0-9]|3[01])$`

**anchor**

If you use an editing platform that relies on an anchor timecode, use Anchor Timecode (Anchor) to specify a timecode that will match the input video frame to the output video frame. Use 24-hour format with frame number, (HH:MM:SS:FF) or (HH:MM:SS;FF). This setting ignores framerate conversion. System behavior for Anchor Timecode varies depending on your setting for Source (TimecodeSource).

* If Source (TimecodeSource) is set to Specified Start (SPECIFIEDSTART), the first input frame is the specified value in Start Timecode (Start). Anchor Timecode (Anchor) and Start Timecode (Start) are used calculate output timecode. * If Source (TimecodeSource) is set to Start at 0 (ZEROBASED) the first frame is 00:00:00:00. * If Source (TimecodeSource) is set to Embedded (EMBEDDED), the first frame is the timecode value on the first input frame of the input.

*Type*: string  
*Required*: False  
*Format*: timecode  
*Pattern*: `^([01][0-9]|2[0-4]):([0-5][0-9]):([0-5][0-9]):([0-5][0-9]):([0-9]{2})$`
Properties

**start**

Only use when you set Source (TimecodeSource) to Specified start (SPECIFIEDSTART). Use Start
timecode (Start) to specify the timecode for the initial frame. Use 24-hour format with frame number,
(HH:MM:SS:FF) or (HH:MM:SS;FF).

Type: string

Required: False

Format: timecode

Pattern: ^([01][0-9]|[2-4]):[0-5][0-9]:[0-5][0-9]:[0-9]{2}|

**source**

Type: string

Required: False

**TimecodeSource (enum)**

Use Source (TimecodeSource) to set how timecodes are handled within this job. To make sure that
your video, audio, captions, and markers are synchronized and that time-based features, such as image
inserter, work correctly, choose the Timecode source option that matches your assets. All timecodes are
in a 24-hour format with frame number (HH:MM:SS:FF).

* Embedded (EMBEDDED) - Use the timecode
that is in the input video. If no embedded timecode is in the source, the service will use Start at 0
(ZEROBASED) instead.

* Start at 0 (ZEROBASED) - Set the timecode of the initial frame to 00:00:00:00.

* Specified Start (SPECIFIEDSTART) - Set the timecode of the initial frame to a value other than zero. You
use Start timecode (Start) to provide this value.

EMBEDDED
ZEROBASED
SPECIFIEDSTART

**TimedMetadata (enum)**

Applies only to HLS outputs. Use this setting to specify whether the service inserts the ID3 timed
metadata from the input in this output.

PASSTHROUGH
NONE

**TimedMetadataInsertion**

**id3Insertions**

Id3Insertions contains the array of Id3Insertion instances.

Type: Array of type Id3Insertion (p. 586)

Required: True

**Timing**

**finishTime**

The time, in Unix epoch format, that the transcoding job finished
submitTime

The time, in Unix epoch format, that you submitted the job.

startTime

The time, in Unix epoch format, that transcoding for the job began.

TtmlDestinationSettings

stylePassthrough

Type: string
Required: False

TtmlStylePassthrough (enum)

Pass through style and position information from a TTML-like input source (TTML, SMPTE-TT, CFF-TT) to the CFF-TT output or TTML output.

- ENABLED
- DISABLED

VideoCodec (enum)

Type of video codec

- FRAME_CAPTURE
- H_264
- H_265
- MPEG2
- PRORES

VideoCodecSettings

h265Settings

Type: H265Settings (p. 570)
Required: False
codec
    Type: string
    Required: True

proresSettings
    Type: ProresSettings (p. 626)
    Required: False

mpeg2Settings
    Type: Mpeg2Settings (p. 614)
    Required: False

frameCaptureSettings
    Type: FrameCaptureSettings (p. 558)
    Required: False

h264Settings
    Type: H264Settings (p. 562)
    Required: False

VideoDescription

fixedAfd
Applies only if you set AFD Signaling(AfdSignaling) to Fixed (FIXED). Use Fixed (FixedAfd) to specify a
four-bit AFD value which the service will write on all frames of this video output.

    Type: integer
    Required: False
    Minimum: 0
    Maximum: 15

scalingBehavior
    Type: string
    Required: False

respondToFafd
    Type: string
    Required: False

codecSettings
    Type: VideoCodecSettings (p. 634)
    Required: True
Properties

afdSignaling
Type: string  
Required: False

colorMetadata
Type: string  
Required: False

timecodeInsertion
Type: string  
Required: False

width
Use Width (Width) to define the video resolution width, in pixels, for this output. If you don't provide a value here, the service will use the input width.

Type: integer  
Required: False  
Minimum: 32  
Maximum: 4096

videoPreprocessors
Find additional transcoding features under Preprocessors (VideoPreprocessors). Enable the features at each output individually. These features are disabled by default.

Type: VideoPreprocessor (p. 637)  
Required: False

antiAlias
Type: string  
Required: False

position
Use Position (Position) to point to a rectangle object to define your position. This setting overrides any other aspect ratio.

Type: Rectangle (p. 628)  
Required: False

sharpness
Use Sharpness (Sharpness) setting to specify the strength of anti-aliasing. This setting changes the width of the anti-alias filter kernel used for scaling. Sharpness only applies if your output resolution is different from your input resolution, and if you set Anti-alias (AntiAlias) to ENABLED. 0 is the softest setting, 100 the sharpest, and 50 recommended for most content.

Type: integer
Required: False
Minimum: 0
Maximum: 100

Crop

Applies only if your input aspect ratio is different from your output aspect ratio. Use Input cropping rectangle (Crop) to specify the video area the service will include in the output. This will crop the input source, causing video pixels to be removed on encode. Do not use this setting if you have enabled Stretch to output (stretchToOutput) in your output settings.

Type: Rectangle (p. 628)
Required: False

dropFrameTimecode

Type: string
Required: False

Height

Use the Height (Height) setting to define the video resolution height for this output. Specify in pixels. If you don’t provide a value here, the service will use the input height.

Type: integer
Required: False
Minimum: 32
Maximum: 2160

VideoDetail

heightInPx

Height in pixels for the output

Type: integer
Required: False

widthInPx

Width in pixels for the output

Type: integer
Required: False

VideoPreprocessor

noiseReducer

Enable the Noise reducer (NoiseReducer) feature to remove noise from your video output if necessary. Enable or disable this feature for each output individually. This setting is disabled by default.

Type: NoiseReducer (p. 620)
Required: False

timecodeBurnin
Timecode burn-in (TimecodeBurnIn)--Burns the output timecode and specified prefix into the output.

  Type: TimecodeBurnin (p. 631)
  Required: False

colorCorrector
Enable the Color corrector (ColorCorrector) feature if necessary. Enable or disable this feature for each output individually. This setting is disabled by default.

  Type: ColorCorrector (p. 540)
  Required: False

deeinterlacer
Use Deinterlacer (Deinterlacer) to produce smoother motion and a clearer picture.

  Type: Deinterlacer (p. 545)
  Required: False

imageInserter
Enable the Image inserter (ImageInserter) feature to include a graphic overlay on your video. Enable or disable this feature for each output individually. This setting is disabled by default.

  Type: ImageInserter (p. 587)
  Required: False

VideoSelector

colorSpace

  Type: string
  Required: False

hdr10Metadata

  Type: Hdr10Metadata (p. 576)
  Required: False

programNumber
Selects a specific program from within a multi-program transport stream. Note that Quad 4K is not currently supported.

  Type: integer
  Required: False
  Minimum: -2147483648
  Maximum: 2147483647
pid

Use PID (Pid) to select specific video data from an input file. Specify this value as an integer; the system automatically converts it to the hexadecimal value. For example, 257 selects PID 0x101. A PID, or packet identifier, is an identifier for a set of data in an MPEG-2 transport stream container.

Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

colorSpaceUsage

Type: string
Required: False

VideoTimecodeInsertion (enum)

Applies only to H.264, H.265, MPEG2, and ProRes outputs. Only enable Timecode insertion when the input framerate is identical to the output framerate. To include timecodes in this output, set Timecode insertion (VideoTimecodeInsertion) to PIC_TIMING_SEI. To leave them out, set it to DISABLED. Default is DISABLED. When the service inserts timecodes in an output, by default, it uses any embedded timecodes from the input. If none are present, the service will set the timecode for the first output frame to zero. To change this default behavior, adjust the settings under Timecode configuration (TimecodeConfig). In the console, these settings are located under Job > Job settings > Timecode configuration. Note - Timecode source under input settings (InputTimecodeSource) does not affect the timecodes that are inserted in the output. Source under Job settings > Timecode configuration (TimecodeSource) does.

DISABLED
PIC_TIMING_SEI

WavFormat (enum)

The service defaults to using RIFF for WAV outputs. If your output audio is likely to exceed 4 GB in file size, or if you otherwise need the extended support of the RF64 format, set your output WAV file format to RF64.

RIFF
RF64

WavSettings

channels

Set Channels to specify the number of channels in this output audio track. With WAV, valid values 1, 2, 4, and 8. In the console, these values are Mono, Stereo, 4-Channel, and 8-Channel, respectively.

Type: integer
Required: False
Minimum: 1
Maximum: 8

bitDepth

Specify Bit depth (BitDepth), in bits per sample, to choose the encoding quality for this audio track.
Presets

URI

/2017-08-29/presets

HTTP Methods

GET

Operation ID: ListPresets

Retrieve a JSON array of up to twenty of your presets. This will return the presets themselves, not just a list of them. To retrieve the next twenty presets, use the nextToken string returned with the array.

Query Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>listBy</td>
<td>String</td>
<td>False</td>
<td></td>
</tr>
<tr>
<td>nextToken</td>
<td>String</td>
<td>False</td>
<td></td>
</tr>
<tr>
<td>maxResults</td>
<td>String</td>
<td>False</td>
<td></td>
</tr>
<tr>
<td>order</td>
<td>String</td>
<td>False</td>
<td></td>
</tr>
</tbody>
</table>

Responses

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>ListPresetsResponse</td>
<td>200 response</td>
</tr>
</tbody>
</table>
### POST

Operation ID: CreatePreset

Create a new preset. For information about job templates see the User Guide at http://docs.aws.amazon.com/mediaconvert/latest/ug/what-is.html

### Responses

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>CreatePresetResponse (p. 652)</td>
<td>201 response</td>
</tr>
<tr>
<td>400</td>
<td>ExceptionBody (p. 663)</td>
<td>BadRequestException</td>
</tr>
<tr>
<td>403</td>
<td>ExceptionBody (p. 663)</td>
<td>AccessDeniedException</td>
</tr>
<tr>
<td>404</td>
<td>ExceptionBody (p. 663)</td>
<td>ResourceNotFoundError</td>
</tr>
<tr>
<td>429</td>
<td>ExceptionBody (p. 663)</td>
<td>LimitExceededException</td>
</tr>
<tr>
<td>409</td>
<td>ExceptionBody (p. 663)</td>
<td>ResourceInUseException</td>
</tr>
</tbody>
</table>

### Schemas

### Request Bodies

#### Example GET

```json
{
  "nextToken (p. 719)": "string",
  "maxResults (p. 719)": integer,
  "listBy (p. 719)": enum,
  "category (p. 719)": "string",
  "order (p. 720)": enum
}
```

#### Example POST

```json
{
  "settings (p. 680)": {
```
"videoDescription (p. 742)": {
  "fixedAfd (p. 750)": integer,
  "scalingBehavior (p. 750)": enum,
  "respondToFad (p. 750)": enum,
  "codecSettings (p. 750)": {
    "h265Settings (p. 749)": {
      "slices (p. 705)": integer,
      "minIInterval (p. 705)": integer,
      "parNumerator (p. 705)": integer,
      "flickerAdaptiveQuantization (p. 705)": enum,
      "gopSizeUnits (p. 705)": enum,
      "hrdBufferSize (p. 706)": integer,
      "qualityTuningLevel (p. 706)": enum,
      "maxBitrate (p. 706)": integer,
      "bitrate (p. 706)": integer,
      "spatialAdaptiveQuantization (p. 706)": enum,
      "sampleAdaptiveOffsetFilterMode (p. 706)": enum,
      "temporalIds (p. 706)": enum,
      "slowPal (p. 706)": enum,
      "tiles (p. 707)": enum,
      "writeMp4PackagingType (p. 707)": enum,
      "codecProfile (p. 707)": enum,
      "alternateTransferFunctionSei (p. 707)": enum,
      "unregisteredSeiTimecode (p. 707)": enum,
      "framerateControl (p. 707)": enum,
      "telecine (p. 707)": enum,
      "codecLevel (p. 707)": enum,
      "framerateConversionAlgorithm (p. 707)": enum,
      "numberReferenceFrames (p. 707)": integer,
      "temporalAdaptiveQuantization (p. 708)": enum,
      "hrdBufferInitialFillPercentage (p. 708)": integer,
      "framerateNumerator (p. 708)": integer,
      "framerateDenominator (p. 708)": integer,
      "gopClosedCadence (p. 708)": integer,
      "numberBFramesBetweenReferenceFrames (p. 708)": integer,
      "framerateDenominator (p. 708)": integer,
      "adaptiveQuantization (p. 709)": enum,
      "interlaceMode (p. 709)": enum,
      "gopSize (p. 709)": number,
      "gopBReference (p. 709)": enum,
      "parNumerator (p. 709)": integer,
      "sceneChangeDetect (p. 709)": enum,
      "parControl (p. 709)": enum,
      "rateControlMode (p. 709)": enum
    },
    "proresSettings (p. 750)": {
      "framerateDenominator (p. 744)": integer,
      "slowPal (p. 744)": enum,
      "framerateControl (p. 744)": enum,
      "telecine (p. 744)": enum,
      "framerateConversionAlgorithm (p. 744)": enum,
      "interlaceMode (p. 744)": enum,
      "parNumerator (p. 745)": integer,
      "codecProfile (p. 745)": enum,
      "parDenominator (p. 745)": integer,
      "parControl (p. 745)": enum,
      "framerateNumerator (p. 745)": integer
    },
    "mpeg2Settings (p. 750)": {
      "minIInterval (p. 734)": integer,
      "parNumerator (p. 735)": integer,
      "gopSizeUnits (p. 735)": enum,
      "hrdBufferSize (p. 735)": integer,
      "qualityTuningLevel (p. 735)": enum,
      "maxBitrate (p. 735)": integer,
      "bitrate (p. 735)": integer,
      "framerateConversionAlgorithm (p. 735)": enum,
      "interlaceMode (p. 735)": enum,
      "parNumerator (p. 736)": integer,
      "parDenominator (p. 736)": integer,
      "parControl (p. 736)": enum,
      "framerateNumerator (p. 736)": integer
    }
  }
}
"spatialAdaptiveQuantization (p. 735)": enum,
"slowPal (p. 736)": enum,
"intraDcPrecision (p. 736)": enum,
"codecProfile (p. 736)": enum,
"softness (p. 736)": integer,
"framerateControl (p. 736)": enum,
"telecine (p. 736)": enum,
"codecLevel (p. 736)": enum,
"framerateConversionAlgorithm (p. 736)": enum,
"temporalAdaptiveQuantization (p. 736)": enum,
"hrdBufferInitialFillPercentage (p. 736)": integer,
"framerateNumerator (p. 737)": integer,
"gopClosedCadence (p. 737)": integer,
"numberBFramesBetweenReferenceFrames (p. 737)": integer,
"framerateDenominator (p. 737)": integer,
"adaptiveQuantization (p. 737)": enum,
"interlaceMode (p. 737)": enum,
"gopSize (p. 738)": number,
"syntax (p. 738)": enum,
"parDenominator (p. 738)": integer,
"sceneChangeDetect (p. 738)": enum,
"parControl (p. 738)": enum,
"rateControlMode (p. 738)": enum
},
"frameCaptureSettings (p. 750)": {
  "framerateDenominator (p. 692)": integer,
  "maxCaptures (p. 693)": integer,
  "framerateNumerator (p. 693)": integer,
  "quality (p. 693)": integer
},
"h264Settings (p. 750)": {
  "slices (p. 696)": integer,
  "minIInterval (p. 696)": integer,
  "parNumerator (p. 697)": integer,
  "flickerAdaptiveQuantization (p. 697)": enum,
  "gopSizeUnits (p. 697)": enum,
  "hrdBufferSize (p. 697)": integer,
  "qualityTuningLevel (p. 697)": enum,
  "maxBitrate (p. 697)": integer,
  "bitrate (p. 697)": integer,
  "adaptiveQuantization (p. 698)": enum,
  "slowPal (p. 698)": enum,
  "codecProfile (p. 698)": enum,
  "unregisteredSeiTimecode (p. 698)": enum,
  "softness (p. 698)": integer,
  "framerateControl (p. 698)": enum,
  "telecine (p. 698)": enum,
  "codecLevel (p. 698)": enum,
  "framerateConversionAlgorithm (p. 698)": enum,
  "numberBFramesBetweenReferenceFrames (p. 699)": integer,
  " temporalAdaptiveQuantization (p. 699)": integer,
  "repeatPps (p. 699)": enum,
  "hrdBufferInitialFillPercentage (p. 699)": integer,
  "framerateNumerator (p. 699)": integer,
  "gopClosedCadence (p. 699)": integer,
  "numberBFramesBetweenReferenceFrames (p. 699)": integer,
  "framerateDenominator (p. 700)": integer,
  "entropyEncoding (p. 700)": enum,
  "fieldEncoding (p. 700)": enum,
  "adaptiveQuantization (p. 700)": enum,
  "interlaceMode (p. 700)": enum,
  "gopSize (p. 700)": number,
  "gopBReference (p. 700)": enum,
  "syntax (p. 700)": enum,
  "parDenominator (p. 701)": integer,
  "sceneChangeDetect (p. 701)": enum,
"parControl (p. 701)": enum,
"rateControlMode (p. 701)": enum
},
"afdSignaling (p. 750)": enum,
"colorMetadata (p. 751)": enum,
"timecodeInsertion (p. 751)": enum,
"width (p. 751)": integer,
"videoPreprocessors (p. 751)": {
"noiseReducer (p. 752)": {
"filter (p. 739)": enum,
"filterSettings (p. 739)": {
"strength (p. 740)": integer
},
"spatialFilterSettings (p. 739)": {
"strength (p. 740)": integer,
"postFilterSharpenStrength (p. 740)": integer,
"speed (p. 740)": integer
}
},
"timecodeBurnin (p. 752)": {
"prefix (p. 748)": "string",
"fontSize (p. 748)": integer,
"position (p. 748)": enum
},
"colorCorrector (p. 752)": {
"saturation (p. 678)": integer,
"brightness (p. 678)": integer,
"hdr10Metadata (p. 678)": {
"redPrimaryY (p. 711)": integer,
"greenPrimaryY (p. 711)": integer,
"whitePointX (p. 711)": integer,
"maxLuminance (p. 711)": integer,
"greenPrimaryX (p. 712)": integer,
"whitePointY (p. 712)": integer,
"redPrimaryX (p. 712)": integer,
"bluePrimaryX (p. 712)": integer,
"bluePrimaryY (p. 712)": integer,
"maxFrameAverageLightLevel (p. 713)": integer,
"maxContentLightLevel (p. 713)": integer,
"minLuminance (p. 713)": integer
},
"contrast (p. 678)": integer,
"hue (p. 679)": integer,
"colorSpaceConversion (p. 679)": enum
},
"deinterlacer (p. 753)": {
"mode (p. 681)": enum,
"control (p. 681)": enum,
"algorithm (p. 681)": enum
},
"imageInserter (p. 753)": {
"insertableImages (p. 713)": [
{
"duration (p. 713)": integer,
"fadeOut (p. 713)": integer,
"imageX (p. 714)": integer,
"fadeIn (p. 714)": integer,
"imageX (p. 714)": integer,
"width (p. 714)": integer,
"startTime (p. 714)": "string",
"opacity (p. 715)": integer,
"layer (p. 715)": integer,
"height (p. 715)": integer,
"imageInserterInput (p. 715)": "string"
}]}
"antiAlias (p. 751)" : enum,
"position (p. 751)" : {
  "width (p. 746)" : integer,
  "x (p. 746)" : integer,
  "y (p. 746)" : integer,
  "height (p. 746)" : integer
},
"sharpness (p. 751)" : integer,
"crop (p. 752)" : {
  "width (p. 746)" : integer,
  "x (p. 746)" : integer,
  "y (p. 746)" : integer,
  "height (p. 746)" : integer
},
"dropFrameTimecode (p. 752)" : enum,
"height (p. 752)" : integer
},
"audioDescriptions (p. 742)" : [
  {
    "audioTypeControl (p. 669)" : enum,
    "languageCodeControl (p. 669)" : enum,
    "remixSettings (p. 669)" : {
      "channelsOut (p. 746)" : integer,
      "channelMapping (p. 746)" : {
        "outputChannels (p. 678)" : [
          {
            "inputChannels (p. 741)" : integer
          }
        ],
        "channelsIn (p. 747)" : integer
      },
      "audioType (p. 670)" : integer,
      "audioSourceName (p. 670)" : "string",
      "codecSettings (p. 670)" : {
        "codec (p. 668)" : enum,
        "wavSettings (p. 669)" : {
          "channels (p. 753)" : integer,
          "bitDepth (p. 753)" : integer,
          "format (p. 754)" : enum,
          "sampleRate (p. 754)" : integer
        },
        "aacSettings (p. 669)" : {
          "vbrQuality (p. 664)" : enum,
          "codecProfile (p. 664)" : enum,
          "codingMode (p. 664)" : enum,
          "specification (p. 664)" : enum,
          "bitrate (p. 663)" : integer,
          "rawFormat (p. 665)" : enum,
          "rateControlMode (p. 665)" : enum,
          "sampleRate (p. 665)" : integer,
          "audioDescriptionBroadcasterMix (p. 665)" : enum
        },
        "ac3Settings (p. 669)" : {
          "dynamicRangeCompressionProfile (p. 666)" : enum,
          "dialognorm (p. 666)" : integer,
          "codingMode (p. 667)" : enum,
          "metadataControl (p. 667)" : enum,
          "bitrate (p. 667)" : integer,
          "lfeFilter (p. 667)" : enum,
          "bitstreamMode (p. 667)" : enum,
"sampleRate (p. 667)": integer,
"aiffSettings (p. 669)": {
  "channels (p. 668)": integer,
  "bitDepth (p. 668)": integer,
  "sampleRate (p. 668)": integer
},
"eac3Settings (p. 669)": {
  "dialnorm (p. 689)": integer,
  "passthroughControl (p. 689)": enum,
  "metadataControl (p. 689)": enum,
  "bitrate (p. 689)": integer,
  "dynamicRangeCompressionRf (p. 689)": enum,
  "sampleRate (p. 689)": integer,
  "surroundEXMode (p. 690)": enum,
  "ltRtSurroundMixLevel (p. 690)": number,
  "dynamicRangeCompressionLine (p. 690)": enum,
  "lfeControl (p. 690)": enum,
  "codingMode (p. 690)": enum,
  "surroundMode (p. 690)": enum,
  "attenuationControl (p. 690)": enum,
  "lfeFilter (p. 690)": enum,
  "phaseControl (p. 690)": enum,
  "ltRtCenterMixLevel (p. 691)": number,
  "dcFilter (p. 691)": enum,
  "bitstreamMode (p. 691)": enum,
  "stereoDownmix (p. 691)": enum,
  "loRoSurroundMixLevel (p. 691)": number,
  "loRoCenterMixLevel (p. 691)": number
},
"mp2Settings (p. 669)": {
  "channels (p. 731)": integer,
  "bitrate (p. 731)": integer,
  "sampleRate (p. 731)": integer
},
"languageCode (p. 670)": enum,
"streamName (p. 670)": "string",
"audioNormalizationSettings (p. 670)": {
  "targetLkfs (p. 672)": number,
  "algorithmControl (p. 672)": enum,
  "loudnessLogging (p. 672)": enum,
  "peakCalculation (p. 672)": enum,
  "correctionGateLevel (p. 672)": integer,
  "algorithm (p. 672)": enum
},
"customLanguageCode (p. 671)": "string"
},
"containerSettings (p. 743)": {
  "container (p. 679)": enum,
  "mp4Settings (p. 679)": {
    "mp4MajorBrand (p. 732)": "string",
    "moovPlacement (p. 732)": enum,
    "csigAtom (p. 732)": enum,
    "freeSpaceBox (p. 732)": enum
  },
  "m3u8Settings (p. 679)": {
    "pmtPid (p. 727)": integer,
    "nielsenId3 (p. 727)": enum,
    "pcrPid (p. 727)": integer,
    "audioPids (p. 728)": [
      integer
    ],
    "audioFramesPerPes (p. 728)": integer,
    "scte35Pid (p. 728)": integer,
    "metadataControl (p. 689)": enum,
    "dynamicRangeCompressionRf (p. 689)": enum,
    "sampleRate (p. 689)": integer,
    "surroundEXMode (p. 690)": enum,
    "ltRtSurroundMixLevel (p. 690)": number,
    "dynamicRangeCompressionLine (p. 690)": enum,
    "lfeControl (p. 690)": enum,
    "codingMode (p. 690)": enum,
    "surroundMode (p. 690)": enum,
    "attenuationControl (p. 690)": enum,
    "lfeFilter (p. 690)": enum,
    "phaseControl (p. 690)": enum,
    "ltRtCenterMixLevel (p. 691)": number,
    "dcFilter (p. 691)": enum,
    "bitstreamMode (p. 691)": enum,
    "stereoDownmix (p. 691)": enum,
    "loRoSurroundMixLevel (p. 691)": number,
    "loRoCenterMixLevel (p. 691)": number
  }
},
"languageCode (p. 670)": enum,
"streamName (p. 670)": "string",
"audioNormalizationSettings (p. 670)": {
  "targetLkfs (p. 672)": number,
  "algorithmControl (p. 672)": enum,
  "loudnessLogging (p. 672)": enum,
  "peakCalculation (p. 672)": enum,
  "correctionGateLevel (p. 672)": integer,
  "algorithm (p. 672)": enum
},
"customLanguageCode (p. 671)": "string"}
"transportStreamId (p. 728)" : integer,
"pcrControl (p. 728)" : enum,
"videoPid (p. 728)" : integer,
"privateMetadataPid (p. 728)" : integer,
"pmtInterval (p. 729)" : integer,
"patInterval (p. 729)" : integer,
"programNumber (p. 729)" : integer,
"timedMetadataPid (p. 729)" : integer,
"timedMetadata (p. 729)" : enum,
"scte35Source (p. 729)" : enum
},
"m2tsSettings (p. 680)" : {
  "dvbTeletextPid (p. 722)" : integer,
  "nielsenId3 (p. 722)" : enum,
  "bitrate (p. 722)" : integer,
  "segmentationTime (p. 722)" : number,
  "audioPids (p. 723)" : [
    integer
  ],
  "rateMode (p. 723)" : enum,
  "maxPcrInterval (p. 723)" : integer,
  "audioFramesPerPes (p. 723)" : integer,
  "ebpAudioInterval (p. 723)" : enum,
  "fragmentTime (p. 723)" : number,
  "scte35Pid (p. 723)" : integer,
  "pmtInterval (p. 724)" : integer,
  "privateMetadataPid (p. 724)" : integer,
  "segmentationStyle (p. 724)" : enum,
  "audioBufferModel (p. 724)" : enum,
  "programNumber (p. 724)" : integer,
  "dvbNitSettings (p. 724)" : {
    "networkName (p. 682)" : "string",
    "networkId (p. 682)" : integer,
    "nitInterval (p. 682)" : integer
  },
  "timedMetadataPid (p. 724)" : integer,
  "scte35Source (p. 725)" : enum,
  "pmtPid (p. 725)" : integer,
  "bufferModel (p. 725)" : enum,
  "ebpPlacement (p. 725)" : enum,
  "dvbSdtSettings (p. 725)" : {
    "sdtInterval (p. 682)" : integer,
    "serviceName (p. 683)" : "string",
    "serviceProviderName (p. 683)" : "string",
    "outputSdt (p. 683)" : enum
  },
  "nullPacketBitrate (p. 725)" : number,
  "pccPId (p. 725)" : integer,
  "minEbpInterval (p. 725)" : integer,
  "transportStreamId (p. 726)" : integer,
  "videoPid (p. 726)" : integer,
  "pcrControl (p. 726)" : enum,
  "esRateInPes (p. 726)" : enum,
  "segmentationMarkers (p. 726)" : enum,
  "dvbTdtSettings (p. 726)" : {
    "tdtInterval (p. 687)" : integer
  },
  "patInterval (p. 726)" : integer,
  "dvbSubPids (p. 727)" : [
    integer
  ],
  "movSettings (p. 680)" : {
    "reference (p. 730)" : enum,
    "paddingControl (p. 730)" : enum,
    "mpeg2FourCCControl (p. 730)" : enum,
"cslgAtom (p. 731)": enum,
"clapAtom (p. 731)": enum,
"f4vSettings (p. 680)": {
  "moovPlacement (p. 692)": enum
},
"captionDescriptions (p. 743)": [
  {
    "languageDescription (p. 676)": "string",
    "destinationSettings (p. 677)": {
      "burninDestinationSettings (p. 677)": {
        "xPosition (p. 673)": integer,
        "backgroundColor (p. 673)": enum,
        "teletextSpacing (p. 673)": enum,
        "yPosition (p. 673)": integer,
        "backgroundOpacity (p. 673)": integer,
        "fontOpacity (p. 673)": integer,
        "fontResolution (p. 674)": integer,
        "shadowOpacity (p. 674)": integer,
        "shadowYOffset (p. 674)": integer,
        "outlineSize (p. 674)": integer,
        "outlineColor (p. 674)": enum,
        "fontSize (p. 674)": integer,
        "shadowXOffset (p. 675)": integer,
        "alignment (p. 675)": integer,
        "shadowColor (p. 675)": enum,
        "fontColor (p. 675)": enum
      },
      "teletextDestinationSettings (p. 677)": {
        "pageNumber (p. 748)": "string"
      },
      "ttmlDestinationSettings (p. 677)": {
        "stylePassthrough (p. 749)": enum
      },
      "destinationType (p. 677)": enum,
      "dvbSubDestinationSettings (p. 677)": {
        "xPosition (p. 683)": integer,
        "backgroundColor (p. 683)": enum,
        "teletextSpacing (p. 683)": enum,
        "yPosition (p. 683)": integer,
        "backgroundOpacity (p. 684)": integer,
        "fontOpacity (p. 684)": integer,
        "fontResolution (p. 684)": integer,
        "shadowOpacity (p. 684)": integer,
        "shadowYOffset (p. 684)": integer,
        "outlineSize (p. 685)": integer,
        "outlineColor (p. 685)": enum,
        "fontSize (p. 685)": integer,
        "shadowXOffset (p. 685)": integer,
        "alignment (p. 685)": enum,
        "shadowColor (p. 685)": enum,
        "fontColor (p. 685)": enum
      },
      "sccDestinationSettings (p. 677)": {
        "framerate (p. 747)": enum
      }
    },
    "languageCode (p. 677)": enum,
    "customLanguageCode (p. 677)": "string"
  }
},
"name (p. 680)": "string",
"description (p. 680)": "string",
"category (p. 681)": "string"
Response Bodies

Example ListPresetsResponse

```json
{
  "presets (p. 720)": [
    {
      "createdAt (p. 741)": "string",
      "lastUpdated (p. 741)": "string",
      "settings (p. 741)": {
        "videoDescription (p. 742)": {
          "fixedAfd (p. 750)": integer,
          "scalingBehavior (p. 750)": enum,
          "respondToAfd (p. 750)": enum,
          "h265Settings (p. 749)": {
            "slices (p. 705)": integer,
            "minInterval (p. 705)": integer,
            "parNumerator (p. 705)": integer,
            "flickerAdaptiveQuantization (p. 705)": enum,
            "gopSizeUnits (p. 705)": enum,
            "hrdBufferSize (p. 706)": integer,
            "qualityTuningLevel (p. 706)": enum,
            "maxBitrate (p. 706)": integer,
            "bitrate (p. 706)": integer,
            "spatialAdaptiveQuantization (p. 706)": enum,
            "sampleAdaptiveOffsetFilterMode (p. 706)": enum,
            "temporalIds (p. 706)": enum,
            "slowPal (p. 706)": enum,
            "tiles (p. 707)": enum,
            "writeMp4PackagingType (p. 707)": enum,
            "codecProfile (p. 707)": enum,
            "alternateTransferFunctionSei (p. 707)": enum,
            "unregisteredSeiTimecode (p. 707)": enum,
            "framerateControl (p. 707)": enum,
            "telecine (p. 707)": enum,
            "codec (p. 749)": enum,
            "proresSettings (p. 750)": {
              "framerateDenominator (p. 744)": integer,
              "slowPal (p. 744)": enum,
              "framerateControl (p. 744)": enum,
              "telecine (p. 744)": enum,
              "framerateConversionAlgorithm (p. 744)": enum,
              "framerateDenominator (p. 709)": integer,
              "gopSize (p. 709)": number,
              "gopBReference (p. 709)": enum,
              "parDenominator (p. 709)": integer,
              "sceneChangeDetect (p. 709)": enum,
              "parControl (p. 709)": enum,
              "rateControlMode (p. 709)": enum
            }
          }
        }
      }
    }
  ]
}
```
"parNumerator (p. 745)": integer,
"codeProfile (p. 745)": enum,
"parDenominator (p. 745)": integer,
"parControl (p. 745)": enum,
"framerateNumerator (p. 745)": integer
},
"mpeg2Settings (p. 750)": {
  "minIInterval (p. 734)": integer,
  "parNumerator (p. 735)": integer,
  "gopSizeUnits (p. 735)": enum,
  "hrdBufferSize (p. 735)": integer,
  "qualityTuningLevel (p. 735)": enum,
  "maxBitrate (p. 735)": integer,
  "bitrate (p. 735)": integer,
  "spatialAdaptiveQuantization (p. 735)": enum,
  "slowPal (p. 736)": enum,
  "intraDcPrecision (p. 736)": enum,
  "codecProfile (p. 736)": enum,
  "softness (p. 736)": integer,
  "framerateControl (p. 736)": enum,
  "telecine (p. 736)": enum,
  "codecLevel (p. 736)": enum,
  "framerateConversionAlgorithm (p. 736)": enum,
  "temporalAdaptiveQuantization (p. 736)": enum,
  "hrdBufferInitialFillPercentage (p. 736)": integer,
  "framerateNumerator (p. 737)": integer,
  "gopClosedCadence (p. 737)": integer,
  "numberBFramesBetweenReferenceFrames (p. 737)": integer,
  "framerateDenominator (p. 737)": integer,
  "adaptiveQuantization (p. 737)": enum,
  "interlaceMode (p. 737)": enum,
  "gopSize (p. 738)": number,
  "syntax (p. 738)": enum,
  "parDenominator (p. 738)": integer,
  "sceneChangeDetect (p. 738)": enum,
  "framerateControl (p. 738)": enum,
  "rateControlMode (p. 738)": enum
},
"frameCaptureSettings (p. 750)": {
  "framerateDenominator (p. 692)": integer,
  "maxCaptures (p. 693)": integer,
  "framerateNumerator (p. 693)": integer,
  "quality (p. 693)": integer
},
"h264Settings (p. 750)": {
  "slices (p. 696)": integer,
  "minIInterval (p. 696)": integer,
  "parNumerator (p. 697)": integer,
  "flickerAdaptiveQuantization (p. 697)": enum,
  "gopSizeUnits (p. 697)": enum,
  "hrdBufferSize (p. 697)": integer,
  "qualityTuningLevel (p. 697)": enum,
  "maxBitrate (p. 697)": integer,
  "bitrate (p. 697)": integer,
  "spatialAdaptiveQuantization (p. 698)": enum,
  "slowPal (p. 698)": enum,
  "codecProfile (p. 698)": enum,
  "unregisteredSeiTimecode (p. 698)": enum,
  "softness (p. 698)": integer,
  "framerateControl (p. 698)": enum,
  "telecine (p. 698)": enum,
  "codecLevel (p. 698)": enum,
  "framerateConversionAlgorithm (p. 698)": enum,
  "numberReferenceFrames (p. 699)": integer,
  "temporalAdaptiveQuantization (p. 699)": enum,
  "repeatPps (p. 699)": enum,
"hrdBufferInitialFillPercentage (p. 699)": integer,
"framerateNumerator (p. 699)": integer,
"gopClosedCadence (p. 699)": integer,
"numberBFramesBetweenReferenceFrames (p. 699)": integer,
"framerateDenominator (p. 700)": integer,
"entropyEncoding (p. 700)": enum,
"fieldEncoding (p. 700)": enum,
"adaptiveQuantization (p. 700)": enum,
"interlaceMode (p. 700)": enum,
"gopSize (p. 700)": number,
"gopBReference (p. 700)": enum,
"syntax (p. 700)": enum,
"parDenominator (p. 701)": integer,
"sceneChangeDetect (p. 701)": enum,
"parControl (p. 701)": enum,
"rateControlMode (p. 701)": enum
"afdSignaling (p. 750)": enum,
"colorMetadata (p. 751)": enum,
"timecodeInsertion (p. 751)": enum,
"width (p. 751)": integer,
"videoPreprocessors (p. 751)": {
  "noiseReducer (p. 752)": {
    "filter (p. 739)": enum,
    "filterSettings (p. 739)": {
      "strength (p. 740)": integer
    },
    "spatialFilterSettings (p. 739)": {
      "strength (p. 740)": integer,
      "postFilterSharpenStrength (p. 740)": integer,
      "speed (p. 740)": integer
    }
  },
  "timecodeBurnin (p. 752)": {
    "prefix (p. 748)": "string",
    "fontSize (p. 748)": integer,
    "position (p. 748)": enum
  },
  "colorCorrector (p. 752)": {
    "saturation (p. 678)": integer,
    "brightness (p. 678)": integer,
    "hdr10Metadata (p. 678)": {
      "redPrimaryY (p. 711)": integer,
      "greenPrimaryY (p. 711)": integer,
      "whitePointX (p. 711)": integer,
      "maxLuminance (p. 711)": integer,
      "greenPrimaryX (p. 712)": integer,
      "whitePointY (p. 712)": integer,
      "redPrimaryX (p. 712)": integer,
      "bluePrimaryX (p. 712)": integer,
      "bluePrimaryY (p. 712)": integer,
      "maxFrameAverageLightLevel (p. 713)": integer,
      "maxContentLightLevel (p. 713)": integer,
      "minLuminance (p. 713)": integer
    },
    "contrast (p. 678)": integer,
    "hue (p. 679)": integer,
    "colorSpaceConversion (p. 679)": enum
  },
  "deinterlacer (p. 753)": {
    "mode (p. 681)": enum,
    "control (p. 681)": enum,
    "algorithm (p. 681)": enum
  },
  "imageInserter (p. 753)": {

"insertableImages (p. 713)": [
  { 
    "duration (p. 713)": integer, 
    "fadeOut (p. 713)": integer, 
    "imageY (p. 714)": integer, 
    "fadeIn (p. 714)": integer, 
    "imageX (p. 714)": integer, 
    "width (p. 714)": integer, 
    "startTime (p. 714)": "string", 
    "opacity (p. 715)": integer, 
    "layer (p. 715)": integer, 
    "height (p. 715)": integer, 
    "imageInserterInput (p. 715)": "string" 
  } 
], 
"antiAlias (p. 751)": enum, 
"position (p. 751)": { 
  "width (p. 746)": integer, 
  "x (p. 746)": integer, 
  "y (p. 746)": integer, 
  "height (p. 746)": integer 
}, 
"sharpness (p. 751)": integer, 
"crop (p. 752)": { 
  "width (p. 746)": integer, 
  "x (p. 746)": integer, 
  "y (p. 746)": integer, 
  "height (p. 746)": integer 
}, 
"dropFrameTimecode (p. 752)": enum, 
"height (p. 752)": integer 
}, 
"audioDescriptions (p. 742)": [ 
  { 
    "audioTypeControl (p. 669)": enum, 
    "languageCodeControl (p. 669)": enum, 
    "remixSettings (p. 669)": { 
      "channelsOut (p. 746)": integer, 
      "channelMapping (p. 746)": { 
        "outputChannels (p. 678)": [ 
          "inputChannels (p. 741)": [ 
            integer 
          ] 
        ] 
      } 
    }, 
    "audioType (p. 670)": integer, 
    "audioSourceName (p. 670)": "string", 
    "codecSettings (p. 670)": { 
      "codec (p. 668)": enum, 
      "wavSettings (p. 669)": { 
        "channels (p. 753)": integer, 
        "bitDepth (p. 753)": integer, 
        "format (p. 754)": enum, 
        "sampleRate (p. 754)": integer 
      }, 
      "aacSettings (p. 669)": { 
        "vbrQuality (p. 664)": enum, 
        "codecProfile (p. 664)": enum, 
        "codingMode (p. 664)": enum, 
        "specification (p. 664)": enum, 
        "..." 
      } 
    } 
  } 
]
"bitrate (p. 665)" : integer,
"rawFormat (p. 665)" : enum,
"rateControlMode (p. 665)" : enum,
"sampleRate (p. 665)" : integer,
"audioDescriptionBroadcasterMix (p. 665)" : enum
},
"ac3Settings (p. 669)" : {
  "dynamicRangeCompressionProfile (p. 666)" : enum,
  "dialnorm (p. 666)" : integer,
  "codingMode (p. 667)" : enum,
  "metadataControl (p. 667)" : enum,
  "bitrate (p. 667)" : integer,
  "lfeFilter (p. 667)" : enum,
  "bitstreamMode (p. 667)" : enum,
  "sampleRate (p. 667)" : integer
},
"aiffSettings (p. 669)" : {
  "channels (p. 668)" : integer,
  "bitDepth (p. 668)" : integer,
  "sampleRate (p. 668)" : integer
},
"eac3Settings (p. 669)" : {
  "dialnorm (p. 689)" : integer,
  "passthroughControl (p. 689)" : enum,
  "metadataControl (p. 689)" : enum,
  "bitrate (p. 689)" : integer,
  "dynamicRangeCompressionRf (p. 689)" : enum,
  "sampleRate (p. 689)" : integer,
  "surroundExMode (p. 690)" : enum,
  "lftRtSurroundMixLevel (p. 690)" : number,
  "dynamicRangeCompressionLine (p. 690)" : enum,
  "lfeControl (p. 690)" : enum,
  "codingMode (p. 690)" : enum,
  "surroundMode (p. 690)" : enum,
  "attenuationControl (p. 690)" : enum,
  "lfeFilter (p. 690)" : enum,
  "phaseControl (p. 690)" : enum,
  "lftRtCenterMixLevel (p. 691)" : number,
  "dcFilter (p. 691)" : enum,
  "bitstreamMode (p. 691)" : enum,
  "stereoDownmix (p. 691)" : enum,
  "loRoSurroundMixLevel (p. 691)" : number,
  "loRoCenterMixLevel (p. 691)" : number
},
"mp2Settings (p. 669)" : {
  "channels (p. 731)" : integer,
  "bitrate (p. 731)" : integer,
  "sampleRate (p. 731)" : integer
},

"languageCode (p. 670)" : enum,
"streamName (p. 670)" : "string",
"audioNormalizationSettings (p. 670)" : {
  "targetLkfs (p. 672)" : number,
  "algorithmControl (p. 672)" : enum,
  "loudnessLogging (p. 672)" : enum,
  "peakCalculation (p. 672)" : enum,
  "correctionGateLevel (p. 672)" : integer,
  "algorithm (p. 672)" : enum
},
"customLanguageCode (p. 671)" : "string"
},

"containerSettings (p. 743)" : {
  "container (p. 679)" : enum,
  "mp4Settings (p. 679)" : {
  "bitrate (p. 762)" : integer,
  "rawFormat (p. 762)" : enum,
  "rateControlMode (p. 762)" : enum,
  "sampleRate (p. 762)" : integer,
  "audioDescriptionBroadcasterMix (p. 762)" : enum
},
"containerSettings (p. 743)" : {
  "container (p. 679)" : enum,
"mp4MajorBrand (p. 732)": "string",
"moovPlacement (p. 732)": enum,
"cslgAtom (p. 732)": enum,
"freeSpaceBox (p. 732)": enum,
"m3u8Settings (p. 679)": {
  "pmtPid (p. 727)": integer,
  "nielsenId3 (p. 727)": enum,
  "pcrPid (p. 727)": integer,
  "audioPids (p. 728)": [integer],
  "audioFramesPerPes (p. 728)": integer,
  "scte35Pid (p. 728)": integer,
  "transportStreamId (p. 728)": integer,
  "pcrControl (p. 728)": enum,
  "videoPid (p. 728)": integer,
  "privateMetadataPid (p. 728)": integer,
  "pmtInterval (p. 729)": integer,
  "patInterval (p. 729)": integer,
  "programNumber (p. 729)": integer,
  "timedMetadataPid (p. 729)": integer,
  "timedMetadata (p. 729)": enum,
  "scte35Source (p. 729)": enum
},
"m2tsSettings (p. 680)": {
  "dvbTeletextPid (p. 722)": integer,
  "nielsenId3 (p. 722)": enum,
  "bitrate (p. 722)": integer,
  "segmentationTime (p. 722)": number,
  "audioPids (p. 723)": [integer],
  "rateMode (p. 723)": enum,
  "maxPcrInterval (p. 723)": integer,
  "audioFramesPerPes (p. 723)": integer,
  "ebpAudioInterval (p. 723)": enum,
  "fragmentTime (p. 723)": number,
  "scte35Pid (p. 723)": integer,
  "pmtInterval (p. 724)": integer,
  "privateMetadataPid (p. 724)": integer,
  "segmentationStyle (p. 724)": enum,
  "audioBufferModel (p. 724)": enum,
  "programNumber (p. 724)": integer,
  "dvbNitSettings (p. 724)": {
    "networkName (p. 682)": "string",
    "networkId (p. 682)": integer,
    "nitInterval (p. 682)": integer
  },
  "timedMetadataPid (p. 724)": integer,
  "scte35Source (p. 725)": enum,
  "pmtPid (p. 725)": integer,
  "bufferModel (p. 725)": enum,
  "ebpPlacement (p. 725)": enum,
  "dvbSdtSettings (p. 725)": {
    "sdtInterval (p. 682)": integer,
    "serviceName (p. 683)": "string",
    "serviceProviderName (p. 683)": "string",
    "outputSdt (p. 683)": enum
  },
  "nullPacketBitrate (p. 725)": number,
  "pcrPid (p. 725)": integer,
  "minEbpInterval (p. 725)": integer,
  "transportStreamId (p. 726)": integer,
  "videoPid (p. 726)": integer,
  "pcrControl (p. 726)": enum,
"esRateInPes (p. 726)": enum,
"segmentationMarkers (p. 726)": enum,
"dvbTdtSettings (p. 726)": {
  "tdtInterval (p. 687)": integer
},
"patInterval (p. 726)": integer,
"dvbSubPdms (p. 727)": [
  integer
],
"movSettings (p. 680)": {
  "reference (p. 730)": enum,
  "paddingControl (p. 730)": enum,
  "mpeg2FourCCControl (p. 730)": enum,
  "cslgAtom (p. 731)": enum,
  "clapAtom (p. 731)": enum
},
"f4vSettings (p. 680)": {
  "moovPlacement (p. 692)": enum
},
"captionDescriptions (p. 743)": [
  {
    "languageDescription (p. 676)": "string",
    "destinationSettings (p. 677)": {
      "burninDestinationSettings (p. 677)": {
        "xPosition (p. 673)": integer,
        "backgroundColor (p. 673)": enum,
        "teletextSpacing (p. 673)": enum,
        "yPosition (p. 673)": integer,
        "backgroundColorOpacity (p. 673)": integer,
        "fontOpacity (p. 673)": integer,
        "fontResolution (p. 674)": integer,
        "shadowOpacity (p. 674)": integer,
        "shadowYOffset (p. 674)": integer,
        "outlineSize (p. 674)": integer,
        "outlineColor (p. 674)": enum,
        "fontSize (p. 674)": integer,
        "shadowXOffset (p. 675)": integer,
        "alignment (p. 675)": enum,
        "shadowColor (p. 675)": enum,
        "fontColor (p. 675)": enum
      },
      "teletextDestinationSettings (p. 677)": {
        "pageNumber (p. 748)": "string"
      },
      "ttmlDestinationSettings (p. 677)": {
        "stylePassthrough (p. 749)": enum
      },
      "destinationType (p. 677)": enum,
      "dvbSubDestinationSettings (p. 677)": {
        "xPosition (p. 683)": integer,
        "backgroundColor (p. 683)": enum,
        "teletextSpacing (p. 683)": enum,
        "yPosition (p. 683)": integer,
        "backgroundColorOpacity (p. 683)": integer,
        "fontOpacity (p. 684)": integer,
        "fontResolution (p. 684)": integer,
        "shadowOpacity (p. 684)": integer,
        "shadowYOffset (p. 684)": integer,
        "outlineSize (p. 685)": integer,
        "outlineColor (p. 685)": enum,
        "fontSize (p. 685)": integer,
        "shadowXOffset (p. 685)": integer,
        "alignment (p. 685)": enum,
        "shadowColor (p. 685)": enum,
"schema (p. 676)": {
  "fontColor (p. 685)": enum,
  "sccDestinationSettings (p. 677)": {
    "framerate (p. 747)": enum
  },
  "languageCode (p. 677)": enum,
  "customLanguageCode (p. 677)": "string"
},

"name (p. 741)": "string",
"description (p. 741)": "string",
"arn (p. 742)": "string",
"category (p. 742)": "string",
"type (p. 742)": enum
],

"nextToken (p. 720)": "string"

---

**Example CreatePresetResponse**

```json
{
  "preset (p. 681)": {
    "createdAt (p. 741)": "string",
    "lastUpdated (p. 741)": "string",
    "settings (p. 741)": {
      "videoDescription (p. 742)": {
        "fixedAfd (p. 750)": integer,
        "scalingBehavior (p. 750)": enum,
        "respondToAfd (p. 750)": enum,
        "codecSettings (p. 750)": {
          "h265Settings (p. 749)": {
            "slices (p. 705)": integer,
            "minInterval (p. 705)": integer,
            "parNumerator (p. 705)": integer,
            "flickerAdaptiveQuantization (p. 705)": enum,
            "gopSizeUnits (p. 705)": enum,
            "hrdBufferSize (p. 706)": integer,
            "qualityTuningLevel (p. 706)": enum,
            "maxBitrate (p. 706)": integer,
            "bitrate (p. 706)": integer,
            "spatialAdaptiveQuantization (p. 706)": enum,
            "sampleAdaptiveOffsetFilterMode (p. 706)": enum,
            "temporalIds (p. 706)": enum,
            "slowPal (p. 706)": enum,
            "tiles (p. 707)": enum,
            "writeMp4PackagingType (p. 707)": enum,
            "codecProfile (p. 707)": enum,
            "alternateTransferFunctionSei (p. 707)": enum,
            "unregisteredSeiTimecode (p. 707)": enum,
            "framerateControl (p. 707)": enum,
            "telecine (p. 707)": enum,
            "codecLevel (p. 707)": enum,
            "framerateConversionAlgorithm (p. 707)": enum,
            "numberReferenceFrames (p. 707)": integer,
            "temporalAdaptiveQuantization (p. 708)": enum,
            "hrdBufferInitialFillPercentage (p. 708)": integer,
            "framerateNumerator (p. 708)": integer,
            "gopClosedCadence (p. 708)": integer,
            "numberBFramesBetweenReferenceFrames (p. 708)": integer,
            "framerateDenominator (p. 708)": integer,
            "adaptiveQuantization (p. 709)": enum,
```
"interlaceMode (p. 709)": enum,
"gopSize (p. 709)": number,
"gopBReference (p. 709)": enum,
"parDenominator (p. 709)": integer,
"sceneChangeDetect (p. 709)": enum,
"parControl (p. 709)": enum,
"rateControlMode (p. 709)": enum
},
"codec (p. 749)": enum,
"proresSettings (p. 750)": {
  "framerateDenominator (p. 744)": integer,
  "slowPal (p. 744)": enum,
  "framerateControl (p. 744)": enum,
  "telecine (p. 744)": enum,
  "framerateConversionAlgorithm (p. 744)": enum,
  "interlaceMode (p. 744)": enum,
  "parNumerator (p. 745)": integer,
  "codecProfile (p. 745)": enum,
  "parDenominator (p. 745)": integer,
  "parControl (p. 745)": enum,
  "framerateNumerator (p. 745)": integer
},
"mpeg2Settings (p. 750)": {
  "minIInterval (p. 734)": integer,
  "parNumerator (p. 735)": integer,
  "gopSizeUnits (p. 735)": enum,
  "hrdBufferSize (p. 735)": integer,
  "qualityTuningLevel (p. 735)": enum,
  "maxBitrate (p. 735)": integer,
  "bitrate (p. 735)": integer,
  "spatialAdaptiveQuantization (p. 735)": enum,
  "slowPal (p. 736)": enum,
  "intraDcPrecision (p. 736)": enum,
  "codecProfile (p. 736)": enum,
  "softness (p. 736)": integer,
  "framerateControl (p. 736)": enum,
  "telecine (p. 736)": enum,
  "codecProfile (p. 736)": enum,
  "softness (p. 736)": integer,
  "framerateControl (p. 736)": enum,
  "telecine (p. 736)": enum,
  "codecLevel (p. 736)": enum,
  "framerateConversionAlgorithm (p. 736)": enum,
  "temporalAdaptiveQuantization (p. 736)": enum,
  "hrdBufferInitialFillPercentage (p. 736)": integer,
  "framerateNumerator (p. 737)": integer,
  "gopClosedCadence (p. 737)": integer,
  "numberBFramesBetweenReferenceFrames (p. 737)": integer,
  "framerateDenominator (p. 737)": integer,
  "adaptiveQuantization (p. 737)": enum,
  "interlaceMode (p. 737)": enum,
  "syntax (p. 738)": number,
  "parDenominator (p. 738)": integer,
  "sceneChangeDetect (p. 738)": enum,
  "parControl (p. 738)": enum,
  "rateControlMode (p. 738)": enum
},
"frameCaptureSettings (p. 750)": {
  "framerateDenominator (p. 692)": integer,
  "maxCaptures (p. 693)": integer,
  "framerateNumerator (p. 693)": integer,
  "quality (p. 693)": integer
},
"h264Settings (p. 750)": {
  "slices (p. 696)": integer,
  "minIInterval (p. 696)": integer,
  "parNumerator (p. 697)": integer,
  "flickerAdaptiveQuantization (p. 697)": enum,
  "gopSizeUnits (p. 697)": enum,
"hrdBufferSize (p. 697)": integer,
"qualityTuningLevel (p. 697)": enum,
"maxBitrate (p. 697)": integer,
"bitrate (p. 697)": integer,
"spatialAdaptiveQuantization (p. 698)": enum,
"slowPal (p. 698)": enum,
"codecProfile (p. 698)": enum,
"unregisteredSeiTimecode (p. 698)": enum,
"softness (p. 698)": integer,
"framerateControl (p. 698)": enum,
"telecine (p. 698)": enum,
"codecLevel (p. 698)": enum,
"framerateConversionAlgorithm (p. 698)": enum,
"numberReferenceFrames (p. 699)": integer,
"temporalAdaptiveQuantization (p. 699)": enum,
"repeatPps (p. 699)": enum,
"hrdBufferInitialFillPercentage (p. 699)": integer,
"framerateNumerator (p. 699)": integer,
"gopClosedCadence (p. 699)": integer,
"numberBFramesBetweenReferenceFrames (p. 699)": integer,
"framerateDenominator (p. 700)": integer,
"entropyEncoding (p. 700)": enum,
"fieldEncoding (p. 700)": enum,
"adaptiveQuantization (p. 700)": enum,
"interlaceMode (p. 700)": enum,
"gopSize (p. 700)": number,
"gopReference (p. 700)": enum,
"syntax (p. 700)": enum,
"parDenominator (p. 701)": integer,
"sceneChangeDetect (p. 701)": enum,
"rateControlMode (p. 701)": enum
},
"afdsignaling (p. 750)": enum,
"colormetadata (p. 751)": enum,
"timecodeinsertion (p. 751)": enum,
"width (p. 751)": integer,
"videoPreprocessors (p. 751)": {
  "noiseReducer (p. 752)": {
    "filter (p. 739)": enum,
    "filterSettings (p. 739)": {
      "strength (p. 740)": integer
    },
    "spatialFilterSettings (p. 739)": {
      "strength (p. 740)": integer,
      "postFilterSharpenStrength (p. 740)": integer,
      "speed (p. 740)": integer
    }
  },
  "timecodeBurnin (p. 752)": {
    "prefix (p. 748)": "string",
    "fontSize (p. 748)": integer,
    "position (p. 748)": enum
  },
  "colorCorrector (p. 752)": {
    "saturation (p. 678)": integer,
    "brightness (p. 678)": integer,
    "hdr10Metadata (p. 678)": {
      "redPrimaryY (p. 711)": integer,
      "greenPrimaryY (p. 711)": integer,
      "whitePointX (p. 711)": integer,
      "maxLuminance (p. 711)": integer,
      "greenPrimaryX (p. 712)": integer,
      "whitePointY (p. 712)": integer,
      "redPrimaryX (p. 712)": integer,
"bluePrimaryX (p. 712)"": integer,
"bluePrimaryY (p. 712)"": integer,
"maxFrameAverageLightLevel (p. 713)"": integer,
"maxContentLightLevel (p. 713)"": integer,
"minLuminance (p. 713)"": integer,
"contrast (p. 678)"": integer,
"hue (p. 679)"": integer,
"colorSpaceConversion (p. 679)"": enum,
"deinterlacer (p. 753)"": {
"mode (p. 681)"": enum,
"control (p. 681)"": enum,
"algorithm (p. 681)"": enum,
"imageInserter (p. 753)"": {
"insertableImages (p. 713)"": [  
{  
"duration (p. 713)"": integer,
"fadeOut (p. 713)"": integer,
"imageY (p. 714)"": integer,
"fadeIn (p. 714)"": integer,
"imageX (p. 714)"": integer,
"width (p. 714)"": integer,
"startTime (p. 714)"": "string",
"opacity (p. 715)"": integer,
"layer (p. 715)"": integer,
"height (p. 715)"": integer,
"imageInserterInput (p. 715)"": "string"
  }
]
},
"antiAlias (p. 751)"": enum,
"position (p. 751)"": {
"width (p. 746)"": integer,
"x (p. 746)"": integer,
"y (p. 746)"": integer,
"height (p. 746)"": integer,
"sharpness (p. 751)"": integer,
"crop (p. 752)"": {
"width (p. 746)"": integer,
"x (p. 746)"": integer,
"y (p. 746)"": integer,
"height (p. 746)"": integer,
"dropFrameTimecode (p. 752)"": enum,
"height (p. 752)"": integer,
"audioDescriptions (p. 742)"": [
{  
"audioTypeControl (p. 669)"": enum,
"languageCodeControl (p. 669)"": enum,
"remixSettings (p. 669)"": {
"channelsOut (p. 746)"": integer,
"channelMapping (p. 746)"": {
"outputChannels (p. 678)"": [  
{  
"inputChannels (p. 741)"": [  
integer
  ]
  }
  ]
}
}
},
"channelsIn (p. 747)"": integer
"audioType (p. 670)": integer,
"audioSourceName (p. 670)": "string",
"codecSettings (p. 670)": {
  "codec (p. 668)": enum,
  "wavSettings (p. 669)": {
    "channels (p. 753)": integer,
    "bitDepth (p. 753)": integer,
    "format (p. 754)": enum,
    "sampleRate (p. 754)": integer
  },
  "aacSettings (p. 669)": {
    "vbrQuality (p. 664)": enum,
    "codecProfile (p. 664)": enum,
    "codingMode (p. 664)": enum,
    "specification (p. 664)": enum,
    "bitrate (p. 665)": integer,
    "rawFormat (p. 665)": enum,
    "rateControlMode (p. 665)": enum,
    "sampleRate (p. 665)": integer,
    "audioDescriptionBroadcasterMix (p. 665)": enum
  },
  "ac3Settings (p. 669)": {
    "dynamicRangeCompressionProfile (p. 666)": enum,
    "dialnorm (p. 666)": integer,
    "codingMode (p. 667)": enum,
    "metadataControl (p. 667)": enum,
    "bitrate (p. 667)": integer,
    "lfeFilter (p. 667)": enum,
    "bitstreamMode (p. 667)": enum,
    "sampleRate (p. 667)": integer
  },
  "siffSettings (p. 669)": {
    "channels (p. 668)": integer,
    "bitDepth (p. 668)": integer,
    "sampleRate (p. 668)": integer
  },
  "eac3Settings (p. 669)": {
    "dialnorm (p. 689)": integer,
    "passthroughControl (p. 689)": enum,
    "metadataControl (p. 689)": enum,
    "bitrate (p. 689)": integer,
    "dynamicRangeCompressionRf (p. 689)": enum,
    "sampleRate (p. 689)": integer,
    "surroundExMode (p. 690)": enum,
    "LtRtSurroundMixLevel (p. 690)": number,
    "dynamicRangeCompressionLine (p. 690)": enum,
    "lifeControl (p. 690)": enum,
    "codingMode (p. 690)": enum,
    "surroundMode (p. 690)": enum,
    "attenuationControl (p. 690)": enum,
    "lfeFilter (p. 690)": enum,
    "phaseControl (p. 690)": enum,
    "LtRtCenterMixLevel (p. 691)": number,
    "dcFilter (p. 691)": enum,
    "bitstreamMode (p. 691)": enum,
    "stereoDownmix (p. 691)": enum,
    "loRoSurroundMixLevel (p. 691)": number,
    "loRoCenterMixLevel (p. 691)": number
  },
  "mp2Settings (p. 669)": {
    "channels (p. 731)": integer,
    "bitrate (p. 731)": integer,
    "sampleRate (p. 731)": integer
  }
},
"eac3Settings (p. 669)": {
  "dialnorm (p. 689)": integer,
  "passthroughControl (p. 689)": enum,
  "metadataControl (p. 689)": enum,
  "bitrate (p. 689)": integer,
  "dynamicRangeCompressionRf (p. 689)": enum,
  "sampleRate (p. 689)": integer,
  "surroundExMode (p. 690)": enum,
  "LtRtSurroundMixLevel (p. 690)": number,
  "dynamicRangeCompressionLine (p. 690)": enum,
  "lifeControl (p. 690)": enum,
  "codingMode (p. 690)": enum,
  "surroundMode (p. 690)": enum,
  "attenuationControl (p. 690)": enum,
  "lfeFilter (p. 690)": enum,
  "phaseControl (p. 690)": enum,
  "LtRtCenterMixLevel (p. 691)": number,
  "dcFilter (p. 691)": enum,
  "bitstreamMode (p. 691)": enum,
  "stereoDownmix (p. 691)": enum,
  "loRoSurroundMixLevel (p. 691)": number,
  "loRoCenterMixLevel (p. 691)": number
},
"languageCode (p. 670)": enum,
"streamName (p. 670)": "string",
"audioNormalizationSettings (p. 670)": {
  "targetLkfs (p. 672)": number,
  "algorithmControl (p. 672)": enum,
  "loudnessLogging (p. 672)": enum,
  "peakCalculation (p. 672)": enum,
  "correctionGateLevel (p. 672)": integer,
  "algorithm (p. 672)": enum
},
"customLanguageCode (p. 671)": "string"
},
"containerSettings (p. 743)": {
  "container (p. 679)": enum,
  "mp4Settings (p. 679)": {
    "mp4MajorBrand (p. 732)": "string",
    "moovPlacement (p. 732)": enum,
    "cslgAtom (p. 732)": enum,
    "freeSpaceBox (p. 732)": enum
  },
  "m3u8Settings (p. 679)": {
    "pmtPid (p. 727)": integer,
    "nielsenId3 (p. 727)": enum,
    "pcrPid (p. 727)": integer,
    "audioPids (p. 728)": [integer]
  },
  "audioFramesPerPes (p. 728)": integer,
  "scte35Pid (p. 728)": integer,
  "transportStreamId (p. 728)": integer,
  "pcrControl (p. 728)": enum,
  "videoPid (p. 728)": integer,
  "privateMetadataPid (p. 728)": integer,
  "pmtInterval (p. 729)": integer,
  "patInterval (p. 729)": integer,
  "programNumber (p. 729)": integer,
  "timedMetaDataPid (p. 729)": integer,
  "timedMetadata (p. 729)": enum,
  "scte35Source (p. 729)": enum
},
"m2tsSettings (p. 680)": {
  "dvbTeletextPid (p. 722)": integer,
  "nielsenId3 (p. 722)": enum,
  "bitrate (p. 722)": integer,
  "segmentationTime (p. 722)": number,
  "audioPids (p. 723)": [integer]
},
"rateMode (p. 723)": enum,
"maxPcrInterval (p. 723)": integer,
"audioFramesPerPes (p. 723)": integer,
"ebpAudioInterval (p. 723)": enum,
"fragmentTime (p. 723)": number,
"scte35Pid (p. 723)": integer,
"pmtInterval (p. 724)": integer,
"privateMetadataPid (p. 724)": integer,
"segmentationStyle (p. 724)": enum,
"audioBufferModel (p. 724)": enum,
"programNumber (p. 724)": integer,
"dvbNitSettings (p. 724)": {
  "networkName (p. 682)": "string",
  "networkId (p. 682)": integer,
  "nitInterval (p. 682)": integer
},
"timedMetadataPid (p. 724)": integer,
"scte35Source (p. 725)": enum,
"pmtPid (p. 725)": integer,
"bufferModel (p. 725)": enum,
"ebpPlacement (p. 725)": enum,
"dvbSdtSettings (p. 725)": {
  "sdtInterval (p. 682)": integer,
  "serviceName (p. 683)": "string",
  "serviceProviderName (p. 683)": "string",
  "outputSdt (p. 683)": enum
},
"nullPacketBitrate (p. 725)": number,
"pcrPid (p. 725)": integer,
"minEbpInterval (p. 725)": integer,
"transportStreamId (p. 726)": integer,
"videoPid (p. 726)": integer,
"pcrControl (p. 726)": enum,
"esRateInPes (p. 726)": enum,
"segmentationMarkers (p. 726)": enum,
"dvbTdtSettings (p. 726)": {
  "tdtInterval (p. 687)": integer
},
"patInterval (p. 726)": integer,
"dvbSubPids (p. 727)": [
  integer
],
"movSettings (p. 680)": {
  "reference (p. 730)": enum,
  "paddingControl (p. 730)": enum,
  "mpeg2FourCCControl (p. 730)": enum,
  "cslgAtom (p. 731)": enum,
  "clapAtom (p. 731)": enum
},
"f4vSettings (p. 680)": {
  "moovPlacement (p. 692)": enum
},
"captionDescriptions (p. 743)": [
  {
    "languageDescription (p. 676)": "string",
    "destinationSettings (p. 677)": {
      "burninDestinationSettings (p. 677)": {
        "xPosition (p. 673)": integer,
        "backgroundColor (p. 673)": enum,
        "teletextSpacing (p. 673)": enum,
        "yPosition (p. 673)": integer,
        "backgroundOpacity (p. 673)": integer,
        "fontOpacity (p. 673)": integer,
        "fontResolution (p. 674)": integer,
        "shadowOpacity (p. 674)": integer,
        "shadowYOffset (p. 674)": integer,
        "outlineSize (p. 674)": integer,
        "outlineColor (p. 674)": enum,
        "fontSize (p. 674)": integer,
        "shadowXOffset (p. 675)": integer,
        "alignment (p. 675)": enum,
        "shadowColor (p. 675)": enum,
        "fontColor (p. 675)": enum
      },
      "teletextDestinationSettings (p. 677)": {
        "pageNumber (p. 748)": "string"
      },
      "ttmlDestinationSettings (p. 677)": {
        "stylePassthrough (p. 749)": enum
      },
      "destinationType (p. 677)": enum
    }
  }
]
"dvbSubDestinationSettings (p. 677)": {
  "xPosition (p. 683)": integer,
  "backgroundColor (p. 683)": enum,
  "teletextSpacing (p. 683)": enum,
  "yPosition (p. 683)": integer,
  "backgroundOpacity (p. 684)": integer,
  "fontOpacity (p. 684)": integer,
  "fontResolution (p. 684)": integer,
  "shadowOpacity (p. 684)": integer,
  "shadowYOffset (p. 684)": integer,
  "outlineSize (p. 685)": integer,
  "outlineColor (p. 685)": enum,
  "fontSize (p. 685)": integer,
  "shadowXOffset (p. 685)": integer,
  "alignment (p. 685)": enum,
  "shadowColor (p. 685)": enum,
  "fontColor (p. 685)": enum
},
"sccDestinationSettings (p. 677)": {
  "framerate (p. 747)": enum
},
"languageCode (p. 677)": enum,
"customLanguageCode (p. 677)": "string"
}

Example ExceptionBody

{
  "message (p. 692)": "string"
}

Properties

**AacAudioDescriptionBroadcasterMix (enum)**

Choose BROADCASTER_MIXED_AD when the input contains pre-mixed main audio + audio description (AD) as a stereo pair. The value for AudioType will be set to 3, which signals to downstream systems that this stream contains “broadcaster mixed AD”. Note that the input received by the encoder must contain pre-mixed audio; the encoder does not perform the mixing. When you choose BROADCASTER_MIXED_AD, the encoder ignores any values you provide in AudioType and FollowInputAudioType. Choose NORMAL when the input does not contain pre-mixed audio + audio description (AD). In this case, the encoder will use any values you provide for AudioType and FollowInputAudioType.

  BROADCASTER_MIXED_AD
  NORMAL

**AacCodecProfile (enum)**

AAC Profile.
AacCodingMode (enum)

Mono (Audio Description), Mono, Stereo, or 5.1 channel layout. Valid values depend on rate control mode and profile. "1.0 - Audio Description (Receiver Mix)" setting receives a stereo description plus control track and emits a mono AAC encode of the description track, with control data emitted in the PES header as per ETSI TS 101 154 Annex E.

- AD_RECEIVER_MIX
- CODING_MODE_1_0
- CODING_MODE_1_1
- CODING_MODE_2_0
- CODING_MODE_5_1

AacRateControlMode (enum)

Rate Control Mode.

- CBR
- VBR

AacRawFormat (enum)

Enables LATM/LOAS AAC output. Note that if you use LATM/LOAS AAC in an output, you must choose "No container" for the output container.

- LATM_LOAS
- NONE

AacSettings

vbrQuality

Type: string

Required: False

codecProfile

Type: string

Required: False

codingMode

Type: string

Required: True

specification

Type: string

Required: False
Properties

**bitrate**

Average bitrate in bits/second. Defaults and valid values depend on rate control mode and profile.

- **Type**: integer
- **Required**: False
- **Minimum**: 6000
- **Maximum**: 1024000

**rawFormat**

- **Type**: string
- **Required**: False

**rateControlMode**

- **Type**: string
- **Required**: False

**sampleRate**

Sample rate in Hz. Valid values depend on rate control mode and profile.

- **Type**: integer
- **Required**: True
- **Minimum**: 8000
- **Maximum**: 96000

**audioDescriptionBroadcasterMix**

- **Type**: string
- **Required**: False

**AacSpecification (enum)**

Use MPEG-2 AAC instead of MPEG-4 AAC audio for raw or MPEG-2 Transport Stream containers.

- MPEG2
- MPEG4

**AacVbrQuality (enum)**

VBR Quality Level - Only used if rate_control_mode is VBR.

- LOW
- MEDIUM_LOW
- MEDIUM_HIGH
- HIGH

**Ac3BitstreamMode (enum)**

Specifies the "Bitstream Mode" (bsmod) for the emitted AC-3 stream. See ATSC A/52-2012 for background on these values.
**COMPLETE_MAIN**
**COMMENTARY**
**DIALOGUE**
**EMERGENCY**
**HEARING_IMPAIRED**
**MUSIC_AND_EFFECTS**
**VISUALLY_IMPAIRED**
**VOICE_OVER**

**Ac3CodingMode (enum)**

Dolby Digital coding mode. Determines number of channels.

- CODING_MODE_1_0
- CODING_MODE_1_1
- CODING_MODE_2_0
- CODING_MODE_3_2_LFE

**Ac3DynamicRangeCompressionProfile (enum)**

If set to FILM_STANDARD, adds dynamic range compression signaling to the output bitstream as defined in the Dolby Digital specification.

- FILM_STANDARD
- NONE

**Ac3LfeFilter (enum)**

Applies a 120Hz lowpass filter to the LFE channel prior to encoding. Only valid with 3_2_LFE coding mode.

- ENABLED
- DISABLED

**Ac3MetadataControl (enum)**

When set to FOLLOW_INPUT, encoder metadata will be sourced from the DD, DD+, or DolbyE decoder that supplied this audio data. If audio was not supplied from one of these streams, then the static metadata settings will be used.

- FOLLOW_INPUT
- USE_CONFIGURED

**Ac3Settings**

**dynamicRangeCompressionProfile**

- **Type**: string
- **Required**: False

**dialnorm**

Sets the dialnorm for the output. If blank and input audio is Dolby Digital, dialnorm will be passed through.
Properties

Type: integer
Required: False
Minimum: 1
Maximum: 31

codingMode
Type: string
Required: False

metadataControl
Type: string
Required: False

bitrate
Average bitrate in bits/second. Valid bitrates depend on the coding mode.
Type: integer
Required: False
Minimum: 64000
Maximum: 640000

lfeFilter
Type: string
Required: False

bitstreamMode
Type: string
Required: False

sampleRate
Sample rate in hz. Sample rate is always 48000.
Type: integer
Required: False
Minimum: 48000
Maximum: 48000

AfdSignaling (enum)
This setting only applies to H.264 and MPEG2 outputs. Use Insert AFD signaling (AfdSignaling) to specify whether the service includes AFD values in the output video data and what those values are. * Choose None to remove all AFD values from this output. * Choose Fixed to ignore input AFD values and instead encode the value specified in the job. * Choose Auto to calculate output AFD values based on the input AFD scaler data.

NONE
AUTO
FIXED
**AiffSettings**

**channels**

Set Channels to specify the number of channels in this output audio track. Choosing Mono in the console will give you 1 output channel; choosing Stereo will give you 2. In the API, valid values are 1 and 2.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 2

**bitDepth**

Specify Bit depth (BitDepth), in bits per sample, to choose the encoding quality for this audio track.

- **Type:** integer
- **Required:** False
- **Minimum:** 16
- **Maximum:** 24

**sampleRate**

Sample rate in hz.

- **Type:** integer
- **Required:** False
- **Minimum:** 8000
- **Maximum:** 192000

**AntiAlias (enum)**

Enable Anti-alias (AntiAlias) to enhance sharp edges in video output when your input resolution is much larger than your output resolution. Default is enabled.

- DISABLED
- ENABLED

**AudioCodec (enum)**

Type of Audio codec.

- AAC
- MP2
- WAV
- AIFF
- AC3
- EAC3
- PASSTHROUGH

**AudioCodecSettings**

**codec**

- **Type:** string
Properties

Required: True

wavSettings

Type: WavSettings (p. 753)
Required: False

aacSettings

Type: AacSettings (p. 664)
Required: False

ac3Settings

Type: Ac3Settings (p. 666)
Required: False

aiffSettings

Type: AiffSettings (p. 668)
Required: False

eac3Settings

Type: Eac3Settings (p. 689)
Required: False

mp2Settings

Type: Mp2Settings (p. 731)
Required: False

AudioDescription

audioTypeControl

Type: string
Required: False

languageCodeControl

Type: string
Required: False

remixSettings

Advanced audio remixing settings.

Type: RemixSettings (p. 746)
Required: False
Properties

audioType

Applies only if Follow Input Audio Type is unchecked (false). A number between 0 and 255. The following are defined in ISO-IEC 13818-1: 0 = Undefined, 1 = Clean Effects, 2 = Hearing Impaired, 3 = Visually Impaired Commentary, 4-255 = Reserved.

Type: integer
Required: False
Minimum: 0
Maximum: 255

audioSourceName

Specifies which audio data to use from each input. In the simplest case, specify an "Audio Selector" by name based on its order within each input. For example if you specify "Audio Selector 3", then the third audio selector will be used from each input. If an input does not have an "Audio Selector 3", then the audio selector marked as "default" in that input will be used. If there is no audio selector marked as "default", silence will be inserted for the duration of that input. Alternatively, an "Audio Selector Group" name may be specified, with similar default/silence behavior. If no audio_source_name is specified, then "Audio Selector 1" will be chosen automatically.

Type: string
Required: False

codecSettings

Type: AudioCodecSettings (p. 668)
Required: True

languageCode

Indicates the language of the audio output track. The ISO 639 language specified in the 'Language Code' drop down will be used when 'Follow Input Language Code' is not selected or when 'Follow Input Language Code' is selected but there is no ISO 639 language code specified by the input.

Type: string
Required: False

streamName

Used for MS Smooth and Apple HLS outputs. Indicates the name displayed by the player (eg. English, or Director Commentary). Alphanumeric characters, spaces, and underscore are legal.

Type: string
Required: False
Pattern: ^[\w\s]*$

audioNormalizationSettings

Type: AudioNormalizationSettings (p. 672)
Required: False
**customLanguageCode**

Specify the language for this audio output track, using the ISO 639-2 or ISO 639-3 three-letter language code. The language specified will be used when 'Follow Input Language Code' is not selected or when 'Follow Input Language Code' is selected but there is no ISO 639 language code specified by the input.

- **Type**: string
- **Required**: False
- **Pattern**: ^[A-Za-z]{3}$

**AudioLanguageCodeControl (enum)**

Choosing FOLLOW_INPUT will cause the ISO 639 language code of the output to follow the ISO 639 language code of the input. The language specified for languageCode' will be used when USE_CONFIGURED is selected or when FOLLOW_INPUT is selected but there is no ISO 639 language code specified by the input.

- FOLLOW_INPUT
- USE_CONFIGURED

**AudioNormalizationAlgorithm (enum)**

Audio normalization algorithm to use. 1770-1 conforms to the CALM Act specification, 1770-2 conforms to the EBU R-128 specification.

- ITU_BS_1770_1
- ITU_BS_1770_2

**AudioNormalizationAlgorithmControl (enum)**

When enabled the output audio is corrected using the chosen algorithm. If disabled, the audio will be measured but not adjusted.

- CORRECT_AUDIO
- MEASURE_ONLY

**AudioNormalizationLoudnessLogging (enum)**

If set to LOG, log each output's audio track loudness to a CSV file.

- LOG
- DONT_LOG

**AudioNormalizationPeakCalculation (enum)**

If set to TRUE_PEAK, calculate and log the TruePeak for each output's audio track loudness.

- TRUE_PEAK
- NONE
**AudioNormalizationSettings**

**targetLkfs**

Target LKFS (loudness) to adjust volume to. If no value is entered, a default value will be used according to the chosen algorithm. The CALM Act (1770-1) recommends a target of -24 LKFS. The EBU R-128 specification (1770-2) recommends a target of -23 LKFS.

- **Type:** number
- **Required:** False
- **Format:** float
- **Minimum:** -59.0
- **Maximum:** 0.0

**algorithmControl**

- **Type:** string
- **Required:** False

**loudnessLogging**

- **Type:** string
- **Required:** False

**peakCalculation**

- **Type:** string
- **Required:** False

**correctionGateLevel**

Content measuring above this level will be corrected to the target level. Content measuring below this level will not be corrected. Gating only applies when not using `real_time_correction`.

- **Type:** integer
- **Required:** False
- **Minimum:** -70
- **Maximum:** 0

**algorithm**

- **Type:** string
- **Required:** False

**AudioTypeControl (enum)**

When set to `FOLLOW_INPUT`, if the input contains an ISO 639 `audio_type`, then that value is passed through to the output. If the input contains no ISO 639 `audio_type`, the value in `Audio Type` is included in the output. Otherwise the value in `Audio Type` is included in the output. Note that this field and `audioType` are both ignored if `audioDescriptionBroadcasterMix` is set to `BROADCASTER_MIXED_AD`.

- `FOLLOW_INPUT`
- `USE_CONFIGURED`
**BurninDestinationSettings**

**xPosition**

Specifies the horizontal position of the caption relative to the left side of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the left of the output. If no explicit x_position is provided, the horizontal caption position will be determined by the alignment parameter. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 2147483647

**backgroundColor**

- **Type**: string
- **Required**: False

**teletextSpacing**

- **Type**: string
- **Required**: False

**yPosition**

Specifies the vertical position of the caption relative to the top of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the top of the output. If no explicit y_position is provided, the caption will be positioned towards the bottom of the output. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 2147483647

**backgroundOpacity**

Specifies the opacity of the background rectangle. 255 is opaque; 0 is transparent. Leaving this parameter blank is equivalent to setting it to 0 (transparent). All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 255

**fontOpacity**

Specifies the opacity of the burned-in captions. 255 is opaque; 0 is transparent. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
Required: True
Minimum: 0
Maximum: 255

fontResolution

Font resolution in DPI (dots per inch); default is 96 dpi. All burn-in and DVB-Sub font settings must match.

Type: integer
Required: False
Minimum: 96
Maximum: 600

shadowOpacity

Specifies the opacity of the shadow. 255 is opaque; 0 is transparent. Leaving this parameter blank is equivalent to setting it to 0 (transparent). All burn-in and DVB-Sub font settings must match.

Type: integer
Required: False
Minimum: 0
Maximum: 255

shadowYOffset

Specifies the vertical offset of the shadow relative to the captions in pixels. A value of -2 would result in a shadow offset 2 pixels above the text. All burn-in and DVB-Sub font settings must match.

Type: integer
Required: False
Minimum: -2147483648
Maximum: 2147483647

outlineSize

Specifies font outline size in pixels. This option is not valid for source captions that are either 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

Type: integer
Required: True
Minimum: 0
Maximum: 10

outlineColor

Type: string
Required: True

fontSize

A positive integer indicates the exact font size in points. Set to 0 for automatic font size selection. All burn-in and DVB-Sub font settings must match.
shadowXOffset

Specifies the horizontal offset of the shadow relative to the captions in pixels. A value of -2 would result in a shadow offset 2 pixels to the left. All burn-in and DVB-Sub font settings must match.

alignment

BurninSubtitleAlignment (enum)

If no explicit x_position or y_position is provided, setting alignment to centered will place the captions at the bottom center of the output. Similarly, setting a left alignment will align captions to the bottom left of the output. If x and y positions are given in conjunction with the alignment parameter, the font will be justified (either left or centered) relative to those coordinates. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

BurninSubtitleBackgroundColor (enum)

Specifies the color of the rectangle behind the captions. All burn-in and DVB-Sub font settings must match.
**BurninSubtitleFontColor (enum)**

Specifies the color of the burned-in captions. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- WHITE
- BLACK
- YELLOW
- RED
- GREEN
- BLUE

**BurninSubtitleOutlineColor (enum)**

Specifies font outline color. This option is not valid for source captions that are either 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- BLACK
- WHITE
- YELLOW
- RED
- GREEN
- BLUE

**BurninSubtitleShadowColor (enum)**

Specifies the color of the shadow cast by the captions. All burn-in and DVB-Sub font settings must match.

- NONE
- BLACK
- WHITE

**BurninSubtitleTeletextSpacing (enum)**

Only applies to jobs with input captions in Teletext or STL formats. Specify whether the spacing between letters in your captions is set by the captions grid or varies depending on letter width. Choose fixed grid to conform to the spacing specified in the captions file more accurately. Choose proportional to make the text easier to read if the captions are closed caption.

- FIXED_GRID
- PROPORTIONAL

**CaptionDescriptionPreset**

**languageDescription**

Human readable information to indicate captions available for players (eg. English, or Spanish). Alphanumeric characters, spaces, and underscore are legal.

- **Type:** string
- **Required:** False
- **Pattern:** `^[\w ]*$`
**destinationSettings**

Type: CaptionDestinationSettings (p. 677)  
Required: True

**languageCode**

Indicates the language of the caption output track.

Type: string  
Required: False

**customLanguageCode**

Indicates the language of the caption output track, using the ISO 639-2 or ISO 639-3 three-letter language code

Type: string  
Required: False  
Pattern: ^[A-Za-z]{3}$

**CaptionDestinationSettings**

**burninDestinationSettings**

Type: BurninDestinationSettings (p. 673)  
Required: False

**teletextDestinationSettings**

Type: TeletextDestinationSettings (p. 748)  
Required: False

**ttmlDestinationSettings**

Type: TtmlDestinationSettings (p. 749)  
Required: False

**destinationType**

Type: string  
Required: True

**dvbSubDestinationSettings**

Type: DvbSubDestinationSettings (p. 683)  
Required: False

**sccDestinationSettings**

Type: SccDestinationSettings (p. 747)  
Required: False
CaptionDestinationType (enum)

Type of Caption output, including Burn-In, Embedded, SCC, SRT, TTML, WebVTT, DVB-Sub, Teletext.

- BURN_IN
- DVB_SUB
- EMBEDDED
- SCC
- SRT
- TELETEXT
- TTML
- WEBVTT

ChannelMapping

outputChannels

List of output channels

- Type: Array of type OutputChannelMapping (p. 741)
- Required: True

ColorCorrector

saturation

Saturation level.

- Type: integer
- Required: False
- Minimum: 1
- Maximum: 100

brightness

Brightness level.

- Type: integer
- Required: False
- Minimum: 1
- Maximum: 100

hdr10Metadata

- Type: Hdr10Metadata (p. 711)
- Required: False

contrast

Contrast level.

- Type: integer
- Required: False
Minimum: 1
Maximum: 100

**hue**

Hue in degrees.

Type: integer
Required: False
Minimum: -180
Maximum: 180

**colorSpaceConversion**

Type: string
Required: False

**ColorMetadata (enum)**

Enable Insert color metadata (ColorMetadata) to include color metadata in this output. This setting is enabled by default.

- IGNORE
- INSERT

**ColorSpaceConversion (enum)**

Determines if colorspace conversion will be performed. If set to _None_, no conversion will be performed. If _Force 601_ or _Force 709_ are selected, conversion will be performed for inputs with differing colorspace. An input's colorspace can be specified explicitly in the "Video Selector":#inputs-video_selector if necessary.

- NONE
- FORCE_601
- FORCE_709
- FORCE_HDR10
- FORCE_HLG_2020

**ContainerSettings**

**container**

Type: string
Required: True

**mp4Settings**

Type: Mp4Settings (p. 732)
Required: False

**m3u8Settings**

Type: M3u8Settings (p. 727)
**Required:** False

**m2tsSettings**

**Type:** M2tsSettings (p. 722)

**Required:** False

**movSettings**

**Type:** MovSettings (p. 730)

**Required:** False

**f4vSettings**

**Type:** F4vSettings (p. 692)

**Required:** False

**ContainerType (enum)**

Container for this output. Some containers require a container settings object. If not specified, the default object will be created.

- F4V
- ISMV
- M2TS
- M3U8
- CMFC
- MOV
- MP4
- MPD
- MXF
- RAW

**CreatePresetRequest**

**settings**

**Type:** PresetSettings (p. 742)

**Required:** True

**name**

The name of the preset you are creating.

**Type:** string

**Required:** True

**description**

Optional. A description of the preset you are creating.

**Type:** string

**Required:** False
category

Optional. A category for the preset you are creating.

- **Type**: string
- **Required**: False

**CreatePresetResponse**

**preset**

- **Type**: Preset (p. 741)
- **Required**: False

**DeinterlaceAlgorithm (enum)**

Only applies when you set Deinterlacer (DeinterlaceMode) to Deinterlace (DEINTERLACE) or Adaptive (ADAPTIVE). Motion adaptive interpolate (INTERPOLATE) produces sharper pictures, while blend (BLEND) produces smoother motion. Use (INTERPOLATE_TICKER) OR (BLEND_TICKER) if your source file includes a ticker, such as a scrolling headline at the bottom of the frame.

- INTERPOLATE
- INTERPOLATE_TICKER
- BLEND
- BLEND_TICKER

**Deinterlacer**

**mode**

- **Type**: string
- **Required**: False

**control**

- **Type**: string
- **Required**: False

**algorithm**

- **Type**: string
- **Required**: False

**DeinterlacerControl (enum)**

- When set to NORMAL (default), the deinterlacer does not convert frames that are tagged in metadata as progressive. It will only convert those that are tagged as some other type. - When set to FORCE_ALL_FRAMES, the deinterlacer converts every frame to progressive - even those that are already tagged as progressive. Turn Force mode on only if there is a good chance that the metadata has tagged frames as progressive when they are not progressive. Do not turn on otherwise; processing frames that are already progressive into progressive will probably result in lower quality video.

- FORCE_ALL_FRAMES
- NORMAL
**DeinterlacerMode (enum)**

Use Deinterlacer (DeinterlaceMode) to choose how the service will do deinterlacing. Default is Deinterlace. - Deinterlace converts interlaced to progressive. - Inverse telecine converts Hard Telecine 29.97i to progressive 23.976p. - Adaptive auto-detects and converts to progressive.

- DEINTERLACE
- INVERSE_TELECINE
- ADAPTIVE

**DropFrameTimecode (enum)**

Applies only to 29.97 fps outputs. When this feature is enabled, the service will use drop-frame timecode on outputs. If it is not possible to use drop-frame timecode, the system will fall back to non-drop-frame. This setting is enabled by default when Timecode insertion (TimecodeInsertion) is enabled.

- DISABLED
- ENABLED

**DvbNitSettings**

**networkName**

The network name text placed in the network_name_descriptor inside the Network Information Table. Maximum length is 256 characters.

- **Type**: string
- **Required**: True

**networkId**

The numeric value placed in the Network Information Table (NIT).

- **Type**: integer
- **Required**: True
- **Minimum**: 0
- **Maximum**: 65535

**nitInterval**

The number of milliseconds between instances of this table in the output transport stream.

- **Type**: integer
- **Required**: True
- **Minimum**: 25
- **Maximum**: 10000

**DvbSdtSettings**

**sdInterval**

The number of milliseconds between instances of this table in the output transport stream.

- **Type**: integer
- **Required**: False
Minimum: 25
Maximum: 2000

**serviceName**

The service name placed in the service_descriptor in the Service Description Table. Maximum length is 256 characters.

*Type: string*
*Required: False*

**serviceProviderName**

The service provider name placed in the service_descriptor in the Service Description Table. Maximum length is 256 characters.

*Type: string*
*Required: False*

**outputSdt**

*Type: string*
*Required: False*

**DvbSubDestinationSettings**

**xPosition**

Specifies the horizontal position of the caption relative to the left side of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the left of the output. If no explicit x_position is provided, the horizontal caption position will be determined by the alignment parameter. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

*Type: integer*
*Required: False*
*Minimum: 0*
*Maximum: 2147483647*

**backgroundColor**

*Type: string*
*Required: False*

**teletextSpacing**

*Type: string*
*Required: False*

**yPosition**

Specifies the vertical position of the caption relative to the top of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the top of the output. If no explicit y_position is
Provided, the caption will be positioned towards the bottom of the output. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

**Type**: integer  
**Required**: False  
**Minimum**: 0  
**Maximum**: 2147483647

### backgroundOpacity

Specifies the opacity of the background rectangle. 255 is opaque; 0 is transparent. Leaving this parameter blank is equivalent to setting it to 0 (transparent). All burn-in and DVB-Sub font settings must match.

**Type**: integer  
**Required**: False  
**Minimum**: 0  
**Maximum**: 255

### fontOpacity

Specifies the opacity of the burned-in captions. 255 is opaque; 0 is transparent. All burn-in and DVB-Sub font settings must match.

**Type**: integer  
**Required**: True  
**Minimum**: 0  
**Maximum**: 255

### fontResolution

Font resolution in DPI (dots per inch); default is 96 dpi. All burn-in and DVB-Sub font settings must match.

**Type**: integer  
**Required**: False  
**Minimum**: 96  
**Maximum**: 600

### shadowOpacity

Specifies the opacity of the shadow. 255 is opaque; 0 is transparent. Leaving this parameter blank is equivalent to setting it to 0 (transparent). All burn-in and DVB-Sub font settings must match.

**Type**: integer  
**Required**: False  
**Minimum**: 0  
**Maximum**: 255

### shadowYOffset

Specifies the vertical offset of the shadow relative to the captions in pixels. A value of -2 would result in a shadow offset 2 pixels above the text. All burn-in and DVB-Sub font settings must match.
Properties

outlineSize

Specifies font outline size in pixels. This option is not valid for source captions that are either 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- Type: integer
- Required: False
- Minimum: -2147483648
- Maximum: 2147483647

outlineColor

- Type: string
- Required: False

fontSize

A positive integer indicates the exact font size in points. Set to 0 for automatic font size selection. All burn-in and DVB-Sub font settings must match.

- Type: integer
- Required: False
- Minimum: 0
- Maximum: 96

shadowXOffset

Specifies the horizontal offset of the shadow relative to the captions in pixels. A value of -2 would result in a shadow offset 2 pixels to the left. All burn-in and DVB-Sub font settings must match.

- Type: integer
- Required: False
- Minimum: -2147483648
- Maximum: 2147483647

alignment

- Type: string
- Required: True

shadowColor

- Type: string
- Required: False

fontColor

- Type: string
**Required**: False

**DvbSubtitleAlignment (enum)**

If no explicit x_position or y_position is provided, setting alignment to centered will place the captions at the bottom center of the output. Similarly, setting a left alignment will align captions to the bottom left of the output. If x and y positions are given in conjunction with the alignment parameter, the font will be justified (either left or centered) relative to those coordinates. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- CENTERED
- LEFT

**DvbSubtitleBackgroundColor (enum)**

Specifies the color of the rectangle behind the captions. All burn-in and DVB-Sub font settings must match.

- NONE
- BLACK
- WHITE

**DvbSubtitleFontColor (enum)**

Specifies the color of the burned-in captions. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- WHITE
- BLACK
- YELLOW
- RED
- GREEN
- BLUE

**DvbSubtitleOutlineColor (enum)**

Specifies font outline color. This option is not valid for source captions that are either 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- BLACK
- WHITE
- YELLOW
- RED
- GREEN
- BLUE

**DvbSubtitleShadowColor (enum)**

Specifies the color of the shadow cast by the captions. All burn-in and DVB-Sub font settings must match.

- NONE
**DvbSubtitleTeletextSpacing (enum)**

Only applies to jobs with input captions in Teletext or STL formats. Specify whether the spacing between letters in your captions is set by the captions grid or varies depending on letter width. Choose fixed grid to conform to the spacing specified in the captions file more accurately. Choose proportional to make the text easier to read if the captions are closed caption.

- FIXED_GRID
- PROPORTIONAL

**DvbTdtSettings**

**tdtInterval**

The number of milliseconds between instances of this table in the output transport stream.

- **Type:** integer
- **Required:** True
- **Minimum:** 1000
- **Maximum:** 30000

**Eac3AttenuationControl (enum)**

If set to ATTENUATE_3_DB, applies a 3 dB attenuation to the surround channels. Only used for 3/2 coding mode.

- ATTENUATE_3_DB
- NONE

**Eac3BitstreamMode (enum)**

Specifies the "Bitstream Mode" (bsmod) for the emitted E-AC-3 stream. See ATSC A/52-2012 (Annex E) for background on these values.

- COMPLETE_MAIN
- COMMENTARY
- EMERGENCY
- HEARING_IMPAIRED
- VISUALLY_IMPAIRED

**Eac3CodingMode (enum)**

Dolby Digital Plus coding mode. Determines number of channels.

- CODING_MODE_1_0
- CODING_MODE_2_0
- CODING_MODE_3_2

**Eac3DcFilter (enum)**

Activates a DC highpass filter for all input channels.
Properties

ENABLED
DISABLED

Eac3DynamicRangeCompressionLine (enum)

Enables Dynamic Range Compression that restricts the absolute peak level for a signal.

NONE
FILM_STANDARD
FILM_LIGHT
MUSIC_STANDARD
MUSIC_LIGHT
SPEECH

Eac3DynamicRangeCompressionRf (enum)

Enables Heavy Dynamic Range Compression, ensures that the instantaneous signal peaks do not exceed specified levels.

NONE
FILM_STANDARD
FILM_LIGHT
MUSIC_STANDARD
MUSIC_LIGHT
SPEECH

Eac3LfeControl (enum)

When encoding 3/2 audio, controls whether the LFE channel is enabled.

LFE
NO_LFE

Eac3LfeFilter (enum)

Applies a 120Hz lowpass filter to the LFE channel prior to encoding. Only valid with 3_2_LFE coding mode.

ENABLED
DISABLED

Eac3MetadataControl (enum)

When set to FOLLOW_INPUT, encoder metadata will be sourced from the DD, DD+, or DolbyE decoder that supplied this audio data. If audio was not supplied from one of these streams, then the static metadata settings will be used.

FOLLOW_INPUT
USE_CONFIGURED

Eac3PassthroughControl (enum)

When set to WHEN_POSSIBLE, input DD+ audio will be passed through if it is present on the input. This detection is dynamic over the life of the transcode. Inputs that alternate between DD+ and non-DD+ content will have a consistent DD+ output as the system alternates between passthrough and encoding.
WHEN_POSSIBLE
NO_PASSTHROUGH

Eac3PhaseControl (enum)
Controls the amount of phase-shift applied to the surround channels. Only used for 3/2 coding mode.
  SHIFT_90_DEGREES
  NO_SHIFT

Eac3Settings

dialnorm
Sets the dialnorm for the output. If blank and input audio is Dolby Digital Plus, dialnorm will be passed through.
  Type: integer
  Required: False
  Minimum: 1
  Maximum: 31

passthroughControl
  Type: string
  Required: False

metadataControl
  Type: string
  Required: False

bitrate
Average bitrate in bits/second. Valid bitrates depend on the coding mode.
  Type: integer
  Required: False
  Minimum: 64000
  Maximum: 640000

dynamicRangeCompressionRf
  Type: string
  Required: False

sampleRate
Sample rate in hz. Sample rate is always 48000.
  Type: integer
  Required: False
**surroundExMode**
- **Type**: string
- **Required**: False

**ltRtSurroundMixLevel**
Left total/Right total surround mix level. Only used for 3/2 coding mode. Valid values: -1.5 -3.0 -4.5 -6.0 -60
- **Type**: number
- **Required**: False
- **Format**: float
- **Minimum**: -60.0
- **Maximum**: -1.5

**dynamicRangeCompressionLine**
- **Type**: string
- **Required**: False

**lfeControl**
- **Type**: string
- **Required**: False

**codingMode**
- **Type**: string
- **Required**: False

**surroundMode**
- **Type**: string
- **Required**: False

**attenuationControl**
- **Type**: string
- **Required**: False

**lfeFilter**
- **Type**: string
- **Required**: False

**phaseControl**
- **Type**: string
**Required**: False

**ltRtCenterMixLevel**

Left total/Right total center mix level. Only used for 3/2 coding mode. Valid values: 3.0, 1.5, 0.0, -1.5, -3.0, -4.5, -6.0, -60

- **Type**: number
- **Required**: False
- **Format**: float
- **Minimum**: -60.0
- **Maximum**: 3.0

**dcFilter**

- **Type**: string
- **Required**: False

**bitstreamMode**

- **Type**: string
- **Required**: False

**stereoDownmix**

- **Type**: string
- **Required**: False

**loRoSurroundMixLevel**

Left only/Right only surround mix level. Only used for 3/2 coding mode. Valid values: -1.5, -3.0, -4.5, -6.0, -60

- **Type**: number
- **Required**: False
- **Format**: float
- **Minimum**: -60.0
- **Maximum**: -1.5

**loRoCenterMixLevel**

Left only/Right only center mix level. Only used for 3/2 coding mode. Valid values: 3.0, 1.5, 0.0, -1.5, -3.0, -4.5, -6.0, -60

- **Type**: number
- **Required**: False
- **Format**: float
- **Minimum**: -60.0
- **Maximum**: 3.0

**Eac3StereoDownmix (enum)**

Stereo downmix preference. Only used for 3/2 coding mode.
Eac3SurroundExMode (enum)

When encoding 3/2 audio, sets whether an extra center back surround channel is matrix encoded into the left and right surround channels.

- NOT_INDICATED
- ENABLED
- DISABLED

Eac3SurroundMode (enum)

When encoding 2/0 audio, sets whether Dolby Surround is matrix encoded into the two channels.

- NOT_INDICATED
- ENABLED
- DISABLED

ExceptionBody

message

Type: string
Required: False

F4vMoovPlacement (enum)

If set to PROGRESSIVE_DOWNLOAD, the MOOV atom is relocated to the beginning of the archive as required for progressive downloading. Otherwise it is placed normally at the end.

- PROGRESSIVE_DOWNLOAD
- NORMAL

F4vSettings

moovPlacement

Type: string
Required: False

FrameCaptureSettings

framerateDenominator

Frame capture will encode the first frame of the output stream, then one frame every framerateDenominator/framerateNumerator seconds. For example, settings of framerateNumerator = 1 and framerateDenominator = 3 (a rate of 1/3 frame per second) will capture the first frame, then 1
frame every 3s. Files will be named as filename.n.jpg where n is the 0-based sequence number of each Capture.

**Type**: integer
**Required**: False
**Minimum**: 1
**Maximum**: 2147483647

**maxCaptures**
Maximum number of captures (encoded jpg output files).

**Type**: integer
**Required**: False
**Minimum**: 1
**Maximum**: 10000000

**framerateNumerator**
Frame capture will encode the first frame of the output stream, then one frame every framerateDenominator/framerateNumerator seconds. For example, settings of framerateNumerator = 1 and framerateDenominator = 3 (a rate of 1/3 frame per second) will capture the first frame, then 1 frame every 3s. Files will be named as filename.NNNNNNN.jpg where N is the 0-based frame sequence number zero padded to 7 decimal places.

**Type**: integer
**Required**: False
**Minimum**: 1
**Maximum**: 2147483647

**quality**
JPEG Quality - a higher value equals higher quality.

**Type**: integer
**Required**: False
**Minimum**: 1
**Maximum**: 100

**H264AdaptiveQuantization (enum)**
Adaptive quantization. Allows intra-frame quantizers to vary to improve visual quality.

- OFF
- LOW
- MEDIUM
- HIGH
- HIGHER
- MAX

**H264CodecLevel (enum)**
H.264 Level.

- AUTO
H264CodecProfile (enum)

H.264 Profile. High 4:2:2 and 10-bit profiles are only available with the AVC-I License.

   BASELINE
   HIGH
   HIGH_10BIT
   HIGH_422
   HIGH_422_10BIT
   MAIN

H264EntropyEncoding (enum)

Entropy encoding mode. Use CABAC (must be in Main or High profile) or CAVLC.

   CABAC
   CAVLC

H264FieldEncoding (enum)

Choosing FORCE_FIELD disables PAFF encoding for interlaced outputs.

   PAFF
   FORCE_FIELD

H264FlickerAdaptiveQuantization (enum)

Adjust quantization within each frame to reduce flicker or 'pop' on I-frames.

   DISABLED
   ENABLED

H264FramerateControl (enum)

If you are using the console, use the Framerate setting to specify the framerate for this output. If you want to keep the same framerate as the input video, choose Follow source. If you want to do framerate conversion, choose a framerate from the dropdown list or choose Custom. The framerates

LEVEL_1
LEVEL_1_1
LEVEL_1_2
LEVEL_1_3
LEVEL_2
LEVEL_2_1
LEVEL_2_2
LEVEL_3
LEVEL_3_1
LEVEL_3_2
LEVEL_4
LEVEL_4_1
LEVEL_4_2
LEVEL_5
LEVEL_5_1
LEVEL_5_2

H264EntropyEncoding (enum)

Entropy encoding mode. Use CABAC (must be in Main or High profile) or CAVLC.

   CABAC
   CAVLC

H264FieldEncoding (enum)

Choosing FORCE_FIELD disables PAFF encoding for interlaced outputs.

   PAFF
   FORCE_FIELD

H264FlickerAdaptiveQuantization (enum)

Adjust quantization within each frame to reduce flicker or 'pop' on I-frames.

   DISABLED
   ENABLED

H264FramerateControl (enum)

If you are using the console, use the Framerate setting to specify the framerate for this output. If you want to keep the same framerate as the input video, choose Follow source. If you want to do framerate conversion, choose a framerate from the dropdown list or choose Custom. The framerates
shown in the dropdown list are decimal approximations of fractions. If you choose Custom, specify your framerate as a fraction. If you are creating your transcoding job specification as a JSON file without the console, use FramerateControl to specify which value the service uses for the framerate for this output. Choose INITIALIZE_FROM_SOURCE if you want the service to use the framerate from the input. Choose SPECIFIED if you want the service to use the framerate you specify in the settings FramerateNumerator and FramerateDenominator.

```
INITIALIZE_FROM_SOURCE
SPECIFIED
```

**H264FramerateConversionAlgorithm (enum)**

When set to INTERPOLATE, produces smoother motion during framerate conversion.

```
DUPLICATE_DROP
INTERPOLATE
```

**H264GopBReference (enum)**

If enable, use reference B frames for GOP structures that have B frames > 1.

```
DISABLED
ENABLED
```

**H264GopSizeUnits (enum)**

Indicates if the GOP Size in H264 is specified in frames or seconds. If seconds the system will convert the GOP Size into a frame count at run time.

```
FRAMES
SECONDS
```

**H264InterlaceMode (enum)**

Use Interlace mode (InterlaceMode) to choose the scan line type for the output. * Top Field First (TOP_FIELD) and Bottom Field First (BOTTOM_FIELD) produce interlaced output with the entire output having the same field polarity (top or bottom first). * Follow, Default Top (FOLLOW_TOP_FIELD) and Follow, Default Bottom (FOLLOW_BOTTOM_FIELD) use the same field polarity as the source. Therefore, behavior depends on the input scan type, as follows. - If the source is interlaced, the output will be interlaced with the same polarity as the source (it will follow the source). The output could therefore be a mix of "top field first" and "bottom field first". - If the source is progressive, the output will be interlaced with "top field first" or "bottom field first" polarity, depending on which of the Follow options you chose.

```
PROGRESSIVE
TOP_FIELD
BOTTOM_FIELD
FOLLOW_TOP_FIELD
FOLLOW_BOTTOM_FIELD
```

**H264ParControl (enum)**

Using the API, enable ParFollowSource if you want the service to use the pixel aspect ratio from the input. Using the console, do this by choosing Follow source for Pixel aspect ratio.

```
INITIALIZE_FROM_SOURCE
```
H264QualityTuningLevel (enum)
Use Quality tuning level (H264QualityTuningLevel) to specify whether to use fast single-pass, high-quality singlepass, or high-quality multipass video encoding.

- SINGLE_PASS
- SINGLE_PASS_HQ
- MULTI_PASS_HQ

H264RateControlMode (enum)
Use this setting to specify whether this output has a variable bitrate (VBR) or constant bitrate (CBR).

- VBR
- CBR

H264RepeatPps (enum)
Places a PPS header on each encoded picture, even if repeated.

- DISABLED
- ENABLED

H264SceneChangeDetect (enum)
Scene change detection (inserts I-frames on scene changes).

- DISABLED
- ENABLED

H264Settings

slices
Number of slices per picture. Must be less than or equal to the number of macroblock rows for progressive pictures, and less than or equal to half the number of macroblock rows for interlaced pictures.

Type: integer
Required: False
Minimum: 1
Maximum: 32

minIInterval
Enforces separation between repeated (cadence) I-frames and I-frames inserted by Scene Change Detection. If a scene change I-frame is within I-interval frames of a cadence I-frame, the GOP is shrunk and/or stretched to the scene change I-frame. GOP stretch requires enabling lookahead as well as setting I-interval. The normal cadence resumes for the next GOP. This setting is only used when Scene Change Detect is enabled. Note: Maximum GOP stretch = GOP size + Min-I-interval - 1

Type: integer
Required: False
Minimum: 0
Maximum: 30

**parNumerator**
Pixel Aspect Ratio numerator.
Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

**flickerAdaptiveQuantization**
Type: string
Required: False

**gopSizeUnits**
Type: string
Required: False

**hrdBufferSize**
Size of buffer (HRD buffer model) in bits. For example, enter five megabits as 5000000.
Type: integer
Required: False
Minimum: 0
Maximum: 1152000000

**qualityTuningLevel**
Type: string
Required: False

**maxBitrate**
Maximum bitrate in bits/second. For example, enter five megabits per second as 5000000.
Type: integer
Required: False
Minimum: 1000
Maximum: 1152000000

**bitrate**
Average bitrate in bits/second. Required for VBR and CBR. For MS Smooth outputs, bitrates must be unique when rounded down to the nearest multiple of 1000.
Type: integer
Required: False
Properties

**spatialAdaptiveQuantization**
- **Type**: string
- **Required**: False

**slowPal**
- **Type**: string
- **Required**: False

**codecProfile**
- **Type**: string
- **Required**: False

**unregisteredSeiTImecode**
- **Type**: string
- **Required**: False

**softness**
Softness. Selects quantizer matrix, larger values reduce high-frequency content in the encoded image.
- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 128

**framerateControl**
- **Type**: string
- **Required**: False

**telecine**
- **Type**: string
- **Required**: False

**codecLevel**
- **Type**: string
- **Required**: False

**framerateConversionAlgorithm**
- **Type**: string
- **Required**: False

Minimum: 1000
Maximum: 1152000000
numberReferenceFrames
Number of reference frames to use. The encoder may use more than requested if using B-frames and/or interlaced encoding.

- Type: integer
- Required: False
- Minimum: 1
- Maximum: 6

temporalAdaptiveQuantization

- Type: string
- Required: False

repeatPps

- Type: string
- Required: False

hrdBufferInitialFillPercentage
Percentage of the buffer that should initially be filled (HRD buffer model).

- Type: integer
- Required: False
- Minimum: 0
- Maximum: 100

framerateNumerator
Framerate numerator - framerate is a fraction, e.g. 24000 / 1001 = 23.976 fps.

- Type: integer
- Required: False
- Minimum: 1
- Maximum: 2147483647

gopClosedCadence
Frequency of closed GOPs. In streaming applications, it is recommended that this be set to 1 so a decoder joining mid-stream will receive an IDR frame as quickly as possible. Setting this value to 0 will break output segmenting.

- Type: integer
- Required: False
- Minimum: 0
- Maximum: 2147483647

numberBFramesBetweenReferenceFrames
Number of B-frames between reference frames.

- Type: integer
- Required: False
framerateDenominator

When you use the API for transcode jobs that use framerate conversion, specify the framerate as a fraction. For example, \( \frac{24000}{1001} = 23.976 \) fps. Use FramerateDenominator to specify the denominator of this fraction. In this example, use 1001 for the value of FramerateDenominator. When you use the console for transcode jobs that use framerate conversion, provide the value as a decimal number for Framerate. In this example, specify 23.976.

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 2147483647

entropyEncoding

- **Type**: string
- **Required**: False

fieldEncoding

- **Type**: string
- **Required**: False

adaptiveQuantization

- **Type**: string
- **Required**: False

interlaceMode

- **Type**: string
- **Required**: False

gopSize

GOP Length (keyframe interval) in frames or seconds. Must be greater than zero.

- **Type**: number
- **Required**: False
- **Format**: float
- **Minimum**: 0.0

gopBReference

- **Type**: string
- **Required**: False

syntax

- **Type**: string
**Required**: False

**parDenominator**

Pixel Aspect Ratio denominator.

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 2147483647

**sceneChangeDetect**

- **Type**: string
- **Required**: False

**parControl**

- **Type**: string
- **Required**: False

**rateControlMode**

- **Type**: string
- **Required**: False

**H264SlowPal (enum)**

Enables Slow PAL rate conversion. 23.976fps and 24fps input is relabeled as 25fps, and audio is sped up correspondingly.

- DISABLED
- ENABLED

**H264SpatialAdaptiveQuantization (enum)**

Adjust quantization within each frame based on spatial variation of content complexity.

- DISABLED
- ENABLED

**H264Syntax (enum)**

Produces a bitstream compliant with SMPTE RP-2027.

- DEFAULT
- RP2027

**H264Telecine (enum)**

This field applies only if the Streams > Advanced > Framerate (framerate) field is set to 29.970. This field works with the Streams > Advanced > Preprocessors > Deinterlacer field (deinterlace_mode) and the Streams > Advanced > Interlaced Mode field (interlace_mode) to identify the scan type for the output:
- Soft: produces 23.976; the player converts this output to 29.97i.

NONE
SOFTWARE HARD

H264TemporalAdaptiveQuantization (enum)
Adjust quantization within each frame based on temporal variation of content complexity.

DISABLED
ENABLED

H264UnregisteredSeiT imecode (enum)
Inserts timecode for each frame as 4 bytes of an unregistered SEI message.

DISABLED
ENABLED

H265AdaptiveQuantization (enum)
Adaptive quantization. Allows intra-frame quantizers to vary to improve visual quality.

OFF
LOW
MEDIUM
HIGH
HIGHER
MAX

H265AlternateTransferFunctionSei (enum)
Enables Alternate Transfer Function SEI message for outputs using Hybrid Log Gamma (HLG) Electro-
Optical Transfer Function (EOTF).

DISABLED
ENABLED

H265CodecLevel (enum)
H.265 Level.

AUTO
LEVEL_1
LEVEL_2
LEVEL_2_1
LEVEL_3
LEVEL_3_1
LEVEL_4
LEVEL_4_1
LEVEL_5
LEVEL_5_1
LEVEL_5_2
LEVEL_6
LEVEL_6_1
LEVEL_6_2

H265CodecProfile (enum)

Represents the Profile and Tier, per the HEVC (H.265) specification. Selections are grouped as [Profile] / [Tier], so "Main/High" represents Main Profile with High Tier. 4:2:2 profiles are only available with the HEVC 4:2:2 License.

MAIN_MAIN
MAIN_HIGH
MAIN10_MAIN
MAIN10_HIGH
MAIN_422_8BIT_MAIN
MAIN_422_8BIT_HIGH
MAIN_422_10BIT_MAIN
MAIN_422_10BIT_HIGH

H265FlickerAdaptiveQuantization (enum)

Adjust quantization within each frame to reduce flicker or 'pop' on I-frames.

DISABLED
ENABLED

H265FramerateControl (enum)

If you are using the console, use the Framerate setting to specify the framerate for this output. If you want to keep the same framerate as the input video, choose Follow source. If you want to do framerate conversion, choose a framerate from the dropdown list or choose Custom. The framerates shown in the dropdown list are decimal approximations of fractions. If you choose Custom, specify your framerate as a fraction. If you are creating your transcoding job specification as a JSON file without the console, use FramerateControl to specify which value the service uses for the framerate for this output. Choose INITIALIZE_FROM_SOURCE if you want the service to use the framerate from the input. Choose SPECIFIED if you want the service to use the framerate you specify in the settings FramerateNumerator and FramerateDenominator.

INITIALIZE_FROM_SOURCE
SPECIFIED

H265FramerateConversionAlgorithm (enum)

When set to INTERPOLATE, produces smoother motion during framerate conversion.

DUPLICATE_DROP
INTERPOLATE

H265GopBReference (enum)

If enable, use reference B frames for GOP structures that have B frames > 1.

DISABLED
ENABLED
H265GopSizeUnits (enum)

Indicates if the GOP Size in H265 is specified in frames or seconds. If seconds the system will convert the
GOP Size into a frame count at run time.

FRAMES
SECONDS

H265InterlaceMode (enum)

Use Interlace mode (InterlaceMode) to choose the scan line type for the output. * Top Field First
(TOP_FIELD) and Bottom Field First (BOTTOM_FIELD) produce interlaced output with the entire output
having the same field polarity (top or bottom first). * Follow, Default Top (FOLLOW_TOP_FIELD) and
Follow, Default Bottom (FOLLOW_BOTTOM_FIELD) use the same field polarity as the source. Therefore,
behavior depends on the input scan type. - If the source is interlaced, the output will be interlaced with
the same polarity as the source (it will follow the source). The output could therefore be a mix of "top
field first" and "bottom field first". - If the source is progressive, the output will be interlaced with "top
field first" or "bottom field first" polarity, depending on which of the Follow options you chose.

PROGRESSIVE
TOP_FIELD
BOTTOM_FIELD
FOLLOW_TOP_FIELD
FOLLOW_BOTTOM_FIELD

H265ParControl (enum)

Using the API, enable ParFollowSource if you want the service to use the pixel aspect ratio from the
input. Using the console, do this by choosing Follow source for Pixel aspect ratio.

INITIALIZE_FROM_SOURCE
SPECIFIED

H265QualityTuningLevel (enum)

Use Quality tuning level (H265QualityTuningLevel) to specify whether to use fast single-pass, high-
quality singlepass, or high-quality multipass video encoding.

SINGLE_PASS
SINGLE_PASS_HQ
MULTI_PASS_HQ

H265RateControlMode (enum)

Use this setting to specify whether this output has a variable bitrate (VBR) or constant bitrate (CBR).

VBR
CBR

H265SampleAdaptiveOffsetFilterMode (enum)

Specify Sample Adaptive Offset (SAO) filter strength. Adaptive mode dynamically selects best strength
based on content

DEFAULT
H265SceneChangeDetect (enum)

Scene change detection (inserts I-frames on scene changes).

   DISABLED
   ENABLED

H265Settings

slices

Number of slices per picture. Must be less than or equal to the number of macroblock rows for progressive pictures, and less than or equal to half the number of macroblock rows for interlaced pictures.

   Type: integer
   Required: False
   Minimum: 1
   Maximum: 32

minIInterval

Enforces separation between repeated (cadence) I-frames and I-frames inserted by Scene Change Detection. If a scene change I-frame is within I-interval frames of a cadence I-frame, the GOP is shrunk and/or stretched to the scene change I-frame. GOP stretch requires enabling lookahead as well as setting I-interval. The normal cadence resumes for the next GOP. This setting is only used when Scene Change Detect is enabled. Note: Maximum GOP stretch = GOP size + Min-I-interval - 1

   Type: integer
   Required: False
   Minimum: 0
   Maximum: 30

parNumerator

Pixel Aspect Ratio numerator.

   Type: integer
   Required: False
   Minimum: 1
   Maximum: 2147483647

flickerAdaptiveQuantization

   Type: string
   Required: False

gopSizeUnits

   Type: string
   Required: False
**hrdBufferSize**

Size of buffer (HRD buffer model) in bits. For example, enter five megabits as 5000000.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 1466400000

**qualityTuningLevel**

- **Type**: string
- **Required**: False

**maxBitrate**

Maximum bitrate in bits/second.

- **Type**: integer
- **Required**: False
- **Minimum**: 1000
- **Maximum**: 1466400000

**bitrate**

Average bitrate in bits/second. Required for VBR and CBR. For MS Smooth outputs, bitrates must be unique when rounded down to the nearest multiple of 1000.

- **Type**: integer
- **Required**: False
- **Minimum**: 1000
- **Maximum**: 1466400000

**spatialAdaptiveQuantization**

- **Type**: string
- **Required**: False

**sampleAdaptiveOffsetFilterMode**

- **Type**: string
- **Required**: False

**temporalIds**

- **Type**: string
- **Required**: False

**slowPal**

- **Type**: string
- **Required**: False
tiles
Type: string
Required: False

writeMp4PackagingType
Type: string
Required: False

codecProfile
Type: string
Required: False

alternateTransferFunctionSei
Type: string
Required: False

unregisteredSeiT imecode
Type: string
Required: False

framerateControl
Type: string
Required: False

telecine
Type: string
Required: False

codecLevel
Type: string
Required: False

framerateConversionAlgorithm
Type: string
Required: False

numberReferenceFrames
Number of reference frames to use. The encoder may use more than requested if using B-frames and/or interlaced encoding.
Type: integer
Required: False
Minimum: 1
Maximum: 6

temporalAdaptiveQuantization
  Type: string
  Required: False

hrdBufferInitialFillPercentage
Percentage of the buffer that should initially be filled (HRD buffer model).
  Type: integer
  Required: False
  Minimum: 0
  Maximum: 100

framerateNumerator
Framerate numerator - framerate is a fraction, e.g. 24000 / 1001 = 23.976 fps.
  Type: integer
  Required: False
  Minimum: 1
  Maximum: 2147483647

gopClosedCadence
Frequency of closed GOPs. In streaming applications, it is recommended that this be set to 1 so a decoder joining mid-stream will receive an IDR frame as quickly as possible. Setting this value to 0 will break output segmenting.
  Type: integer
  Required: False
  Minimum: 0
  Maximum: 2147483647

numberBFramesBetweenReferenceFrames
Number of B-frames between reference frames.
  Type: integer
  Required: False
  Minimum: 0
  Maximum: 7

framerateDenominator
Framerate denominator.
  Type: integer
  Required: False
  Minimum: 1
  Maximum: 2147483647
Properties

adaptiveQuantization
Type: string  
Required: False

interlaceMode
Type: string  
Required: False

gopSize
GOP Length (keyframe interval) in frames or seconds. Must be greater than zero.
Type: number  
Required: False  
Format: float  
Minimum: 0.0

gopBReference
Type: string  
Required: False

parDenominator
Pixel Aspect Ratio denominator.
Type: integer  
Required: False  
Minimum: 1  
Maximum: 2147483647

sceneChangeDetect
Type: string  
Required: False

parControl
Type: string  
Required: False

rateControlMode
Type: string  
Required: False

H265SlowPal (enum)
Enables Slow PAL rate conversion. 23.976fps and 24fps input is relabeled as 25fps, and audio is sped up correspondingly.
DISABLED
ENABLED

**H265SpatialAdaptiveQuantization (enum)**

Adjust quantization within each frame based on spatial variation of content complexity.

DISABLED
ENABLED

**H265Telecine (enum)**

This field applies only if the Streams > Advanced > Framerate (framerate) field is set to 29.970. This field works with the Streams > Advanced > Preprocessors > Deinterlacer field (deinterlace_mode) and the Streams > Advanced > Interlaced Mode field (interlace_mode) to identify the scan type for the output: Progressive, Interlaced, Hard Telecine or Soft Telecine. - Hard: produces 29.97i output from 23.976 input. - Soft: produces 23.976; the player converts this output to 29.97i.

NONE
SOFT
HARD

**H265TemporalAdaptiveQuantization (enum)**

Adjust quantization within each frame based on temporal variation of content complexity.

DISABLED
ENABLED

**H265TemporalIds (enum)**

Enables temporal layer identifiers in the encoded bitstream. Up to 3 layers are supported depending on GOP structure: I- and P-frames form one layer, reference B-frames can form a second layer and non-reference b-frames can form a third layer. Decoders can optionally decode only the lower temporal layers to generate a lower frame rate output. For example, given a bitstream with temporal IDs and with b-frames = 1 (i.e. IbpBpb display order), a decoder could decode all the frames for full frame rate output or only the I and P frames (lowest temporal layer) for a half frame rate output.

DISABLED
ENABLED

**H265Tiles (enum)**

Enable use of tiles, allowing horizontal as well as vertical subdivision of the encoded pictures.

DISABLED
ENABLED

**H265UnregisteredSeiTimecode (enum)**

Inserts timecode for each frame as 4 bytes of an unregistered SEI message.

DISABLED
ENABLED
H265WriteMp4PackagingType (enum)

If HVC1, output that is H.265 will be marked as HVC1 and adhere to the ISO-IECJTC1-SC29_N13798_Text_ISOIEC_FDIS_14496-15_3rd_E spec which states that parameter set NAL units will be stored in the sample headers but not in the samples directly. If HEV1, then H.265 will be marked as HEV1 and parameter set NAL units will be written into the samples.

- HVC1
- HEV1

Hdr10Metadata

redPrimaryY

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

- Type: integer
- Required: False
- Minimum: 0
- Maximum: 50000

greenPrimaryY

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

- Type: integer
- Required: False
- Minimum: 0
- Maximum: 50000

whitePointX

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

- Type: integer
- Required: False
- Minimum: 0
- Maximum: 50000

maxLuminance

Nominal maximum mastering display luminance in units of 0.0001 candelas per square meter.

- Type: integer
- Required: False
- Minimum: 0
- Maximum: 2147483647
greenPrimaryX

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

Type: integer  
Required: False  
Minimum: 0  
Maximum: 50000

whitePointY

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

Type: integer  
Required: False  
Minimum: 0  
Maximum: 50000

redPrimaryX

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

Type: integer  
Required: False  
Minimum: 0  
Maximum: 50000

bluePrimaryX

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

Type: integer  
Required: False  
Minimum: 0  
Maximum: 50000

bluePrimaryY

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

Type: integer  
Required: False  
Minimum: 0  
Maximum: 50000
maxFrameAverageLightLevel

Maximum average light level of any frame in the coded video sequence, in units of candelas per square meter.

Type: integer  
Required: True  
Minimum: 0  
Maximum: 65535

maxContentLightLevel

Maximum light level among all samples in the coded video sequence, in units of candelas per square meter.

Type: integer  
Required: True  
Minimum: 0  
Maximum: 65535

minLuminance

Nominal minimum mastering display luminance in units of 0.0001 candelas per square meter.

Type: integer  
Required: False  
Minimum: 0  
Maximum: 2147483647

ImageInserter

insertableImages

Image to insert. Must be 32 bit windows BMP, PNG, or TGA file. Must not be larger than the output frames.

Type: Array of type InsertableImage (p. 713)  
Required: True

InsertableImage

duration

Use Duration (Duration) to set the time, in milliseconds, for the image to remain on the output video.

Type: integer  
Required: False  
Minimum: -2147483648  
Maximum: 2147483647

fadeOut

Use Fade out (FadeOut) to set the length, in milliseconds, of the inserted image fade out. If you don’t specify a value for Fade out, the image will disappear abruptly at the end of the inserted image duration.
Type: integer  
Required: False  
Minimum: -2147483648  
Maximum: 2147483647

**imageY**

Use Top (ImageY) to set the distance, in pixels, between the inserted image and the top edge of the video frame. Required for BMP, PNG and TGA input.

Type: integer  
Required: True  
Minimum: -2147483648  
Maximum: 2147483647

**fadeIn**

Use Fade in (FadeIut) to set the length, in milliseconds, of the inserted image fade in. If you don't specify a value for Fade in, the image will appear abruptly at the Start time.

Type: integer  
Required: False  
Minimum: -2147483648  
Maximum: 2147483647

**imageX**

Use Left (ImageX) to set the distance, in pixels, between the inserted image and the left edge of the frame. Required for BMP, PNG and TGA input.

Type: integer  
Required: True  
Minimum: -2147483648  
Maximum: 2147483647

**width**

Specify the Width (Width) of the inserted image. Use a value that is less than or equal to the video resolution width. Leave this setting blank to use the native width of the image.

Type: integer  
Required: False  
Minimum: -2147483648  
Maximum: 2147483647

**startTime**

Use Start time (StartTime) to specify the video timecode when the image is inserted in the output. This must be in timecode (HH:MM:SS:FF or HH:MM:SS;FF) format.

Type: string  
Required: False  
Pattern: ^((([0-1]\d)|(2[0-3]))(:([0-5]\d)\2))([;\,;][0-5]\d)$
opacity
Use Opacity (Opacity) to specify how much of the underlying video shows through the inserted image. 0 is transparent and 100 is fully opaque. Default is 50.

  Type: integer
  Required: True
  Minimum: 0
  Maximum: 100

layer
Use Layer (Layer) to specify how overlapping inserted images appear. Images with higher values of layer appear on top of images with lower values of layer.

  Type: integer
  Required: True
  Minimum: 0
  Maximum: 99

height
Specify the Height (Height) of the inserted image. Use a value that is less than or equal to the video resolution height. Leave this setting blank to use the native height of the image.

  Type: integer
  Required: False
  Minimum: -2147483648
  Maximum: 2147483647

imageInserterInput
Use Image location (imageInserterInput) to specify the Amazon S3 location of the image to be inserted into the output. Use a 32 bit BMP, PNG, or TGA file that fits inside the video frame.

  Type: string
  Required: True
  Pattern: ^s3://.*\.(bmp|BMP|png|PNG|tga|TGA)$

LanguageCode (enum)

  ENG
  SPA
  FRA
  DEU
  GER
  ZHO
  ARA
  HIN
  JPN
  RUS
ListPresetsRequest

nextToken

Use this string, provided with the response to a previous request, to request the next batch of presets.

Type: string  
Required: False

maxResults

Optional. Number of presets, up to twenty, that will be returned at one time

Type: integer
Required: False
Format: int32
Minimum: 1
Maximum: 20

listBy

Type: string
Required: False

category

Optionally, specify a preset category to limit responses to only presets from that category.

Type: string
Required: False
order
  Type: string
  Required: False

ListPresetsResponse

presets
List of presets
  Type: Array of type Preset (p. 741)
  Required: False

nextToken
Use this string to request the next batch of presets.
  Type: string
  Required: False

M2tsAudioBufferModel (enum)
Selects between the DVB and ATSC buffer models for Dolby Digital audio.
  DVB
  ATSC

M2tsBufferModel (enum)
Controls what buffer model to use for accurate interleaving. If set to MULTIPLEX, use multiplex buffer model. If set to NONE, this can lead to lower latency, but low-memory devices may not be able to play back the stream without interruptions.
  MULTIPLEX
  NONE

M2tsEbpAudioInterval (enum)
When set to VIDEO_AND_FIXED_INTERVALS, audio EBP markers will be added to partitions 3 and 4. The interval between these additional markers will be fixed, and will be slightly shorter than the video EBP marker interval. When set to VIDEO_INTERVAL, these additional markers will not be inserted. Only applicable when EBP segmentation markers are is selected (segmentationMarkers is EBP or EBP_LEGACY).
  VIDEO_AND_FIXED_INTERVALS
  VIDEO_INTERVAL

M2tsEbpPlacement (enum)
Selects which PIDs to place EBP markers on. They can either be placed only on the video PID, or on both the video PID and all audio PIDs. Only applicable when EBP segmentation markers are is selected (segmentationMarkers is EBP or EBP_LEGACY).
VIDEO_AND_AUDIO_PIDS
VIDEO_PID

**M2tsEsRateInPes (enum)**

Controls whether to include the ES Rate field in the PES header.

- INCLUDE
- EXCLUDE

**M2tsNielsenId3 (enum)**

If INSERT, Nielsen inaudible tones for media tracking will be detected in the input audio and an equivalent ID3 tag will be inserted in the output.

- INSERT
- NONE

**M2tsPcrControl (enum)**

When set to PCR_EVERY_PES_PACKET, a Program Clock Reference value is inserted for every Packetized Elementary Stream (PES) header. This is effective only when the PCR PID is the same as the video or audio elementary stream.

- PCR_EVERY_PES_PACKET
- CONFIGURED_PCR_PERIOD

**M2tsRateMode (enum)**

When set to CBR, inserts null packets into transport stream to fill specified bitrate. When set to VBR, the bitrate setting acts as the maximum bitrate, but the output will not be padded up to that bitrate.

- VBR
- CBR

**M2tsScte35Source (enum)**

Enables SCTE-35 passthrough (scte35Source) to pass any SCTE-35 signals from input to output.

- PASSTHROUGH
- NONE

**M2tsSegmentationMarkers (enum)**

Inserts segmentation markers at each segmentation_time period. rai_segstart sets the Random Access Indicator bit in the adaptation field. rai_adapt sets the RAI bit and adds the current timecode in the private data bytes. psi_segstart inserts PAT and PMT tables at the start of segments. ebp adds Encoder Boundary Point information to the adaptation field as per OpenCable specification OC-SP-EBP-I01-130118. ebp_legacy adds Encoder Boundary Point information to the adaptation field using a legacy proprietary format.

- NONE
- RAI_SEGSTART
- RAI_ADAPT
M2tsSegmentationStyle (enum)

The segmentation style parameter controls how segmentation markers are inserted into the transport stream. With avails, it is possible that segments may be truncated, which can influence where future segmentation markers are inserted. When a segmentation style of "reset_cadence" is selected and a segment is truncated due to an avail, we will reset the segmentation cadence. This means the subsequent segment will have a duration of \( \frac{\text{segmentation_time}}{\text{segmentation_time}} \) seconds. When a segmentation style of "maintain_cadence" is selected and a segment is truncated due to an avail, we will not reset the segmentation cadence. This means the subsequent segment will likely be truncated as well. However, all segments after that will have a duration of \( \frac{\text{segmentation_time}}{\text{segmentation_time}} \) seconds. Note that EBP lookahead is a slight exception to this rule.

- MAINTAIN_CADENCE
- RESET_CADENCE

M2tsSettings

dvbTeletextPid

Packet Identifier (PID) for input source DVB Teletext data to this output.

- **Type**: integer
- **Required**: False
- **Minimum**: 32
- **Maximum**: 8182

nielsenId3

- **Type**: string
- **Required**: False

bitrate

The output bitrate of the transport stream in bits per second. Setting to 0 lets the muxer automatically determine the appropriate bitrate. Other common values are 3750000, 7500000, and 15000000.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 2147483647

segmentationTime

The length in seconds of each segment. Required unless markers is set to _none_.

- **Type**: number
- **Required**: False
- **Format**: float
- **Minimum**: 0.0
**audioPids**

Packet Identifier (PID) of the elementary audio stream(s) in the transport stream. Multiple values are accepted, and can be entered in ranges and/or by comma separation.

- **Type**: Array of type integer
- **Required**: False

**rateMode**

- **Type**: string
- **Required**: False

**maxPcrInterval**

Maximum time in milliseconds between Program Clock References (PCRs) inserted into the transport stream.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 500

**audioFramesPerPes**

The number of audio frames to insert for each PES packet.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 2147483647

**ebpAudioInterval**

- **Type**: string
- **Required**: False

**fragmentTime**

The length in seconds of each fragment. Only used with EBP markers.

- **Type**: number
- **Required**: False
- **Format**: float
- **Minimum**: 0.0

**scte35Pid**

Packet Identifier (PID) of the SCTE-35 stream in the transport stream.

- **Type**: integer
- **Required**: False
- **Minimum**: 32
Maximum: 8182

**pmtInterval**

The number of milliseconds between instances of this table in the output transport stream.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 1000

**privateMetadataPid**

Packet Identifier (PID) of the private metadata stream in the transport stream.

- **Type**: integer
- **Required**: False
- **Minimum**: 32
- **Maximum**: 8182

**segmentationStyle**

- **Type**: string
- **Required**: False

**audioBufferModel**

- **Type**: string
- **Required**: False

**programNumber**

The value of the program number field in the Program Map Table.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 65535

**dvbNitSettings**

- **Type**: DvbNitSettings (p. 682)
- **Required**: False

**timedMetadataPid**

Packet Identifier (PID) of the timed metadata stream in the transport stream.

- **Type**: integer
- **Required**: False
- **Minimum**: 32
- **Maximum**: 8182
**scte35Source**

Type: string
Required: False

**pmtPid**

Packet Identifier (PID) for the Program Map Table (PMT) in the transport stream.

Type: integer
Required: False
Minimum: 32
Maximum: 8182

**bufferModel**

Type: string
Required: False

**ebpPlacement**

Type: string
Required: False

**dvbSdtSettings**

Type: DvbSdtSettings (p. 682)
Required: False

**nullPacketBitrate**

Value in bits per second of extra null packets to insert into the transport stream. This can be used if a downstream encryption system requires periodic null packets.

Type: number
Required: False
Format: float
Minimum: 0.0

**pcrPid**

Packet Identifier (PID) of the Program Clock Reference (PCR) in the transport stream. When no value is given, the encoder will assign the same value as the Video PID.

Type: integer
Required: False
Minimum: 32
Maximum: 8182

**minEbpInterval**

When set, enforces that Encoder Boundary Points do not come within the specified time interval of each other by looking ahead at input video. If another EBP is going to come in within the specified time interval, the current EBP is not emitted, and the segment is "stretched" to the next marker. The
lookahead value does not add latency to the system. The Live Event must be configured elsewhere to create sufficient latency to make the lookahead accurate.

- **Type**: integer
  - **Required**: False
  - **Minimum**: 0
  - **Maximum**: 10000

### transportStreamId

The value of the transport stream ID field in the Program Map Table.

- **Type**: integer
  - **Required**: False
  - **Minimum**: 0
  - **Maximum**: 65535

### videoPid

Packet Identifier (PID) of the elementary video stream in the transport stream.

- **Type**: integer
  - **Required**: False
  - **Minimum**: 32
  - **Maximum**: 8182

### pcrControl

- **Type**: string
  - **Required**: False

### esRateInPes

- **Type**: string
  - **Required**: False

### segmentationMarkers

- **Type**: string
  - **Required**: False

### dvbTdtSettings

- **Type**: [DvbTdtSettings](p. 687)
  - **Required**: False

### patInterval

The number of milliseconds between instances of this table in the output transport stream.

- **Type**: integer
  - **Required**: False
  - **Minimum**: 0
  - **Maximum**: 1000
**dvbSubPids**
Packet Identifier (PID) for input source DVB Subtitle data to this output. Multiple values are accepted, and can be entered in ranges and/or by comma separation.

- **Type:** Array of type integer
- **Required:** False

**M3u8NielsenId3 (enum)**
If INSERT, Nielsen inaudible tones for media tracking will be detected in the input audio and an equivalent ID3 tag will be inserted in the output.

- **INSERT**
- **NONE**

**M3u8PcrControl (enum)**
When set to PCR_EVERY_PES_PACKET a Program Clock Reference value is inserted for every Packetized Elementary Stream (PES) header. This parameter is effective only when the PCR PID is the same as the video or audio elementary stream.

- **PCR_EVERY_PES_PACKET**
- **CONFIGURED_PCR_PERIOD**

**M3u8Scte35Source (enum)**
Enables SCTE-35 passthrough (scte35Source) to pass any SCTE-35 signals from input to output.

- **PASSTHROUGH**
- **NONE**

**M3u8Settings**

**pmtPid**
Packet Identifier (PID) for the Program Map Table (PMT) in the transport stream.

- **Type:** integer
- **Required:** False
- **Minimum:** 32
- **Maximum:** 8182

**nielsenId3**
- **Type:** string
- **Required:** False

**pcrPid**
Packet Identifier (PID) of the Program Clock Reference (PCR) in the transport stream. When no value is given, the encoder will assign the same value as the Video PID.

- **Type:** integer
- **Required:** False
<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Required</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>integer</td>
<td>False</td>
<td>32</td>
<td>8182</td>
</tr>
<tr>
<td>Maximum</td>
<td>integer</td>
<td>False</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**audioPids**

Packet Identifier (PID) of the elementary audio stream(s) in the transport stream. Multiple values are accepted, and can be entered in ranges and/or by comma separation.

- **Type**: Array of type integer
- **Required**: False

**audioFramesPerPes**

The number of audio frames to insert for each PES packet.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 2147483647

**scte35Pid**

Packet Identifier (PID) of the SCTE-35 stream in the transport stream.

- **Type**: integer
- **Required**: False
- **Minimum**: 32
- **Maximum**: 8182

**transportStreamId**

The value of the transport stream ID field in the Program Map Table.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 65535

**pcrControl**

- **Type**: string
- **Required**: False

**videoPid**

Packet Identifier (PID) of the elementary video stream in the transport stream.

- **Type**: integer
- **Required**: False
- **Minimum**: 32
- **Maximum**: 8182

**privateMetadataPid**

Packet Identifier (PID) of the private metadata stream in the transport stream.
Properties

pmtInterval
The number of milliseconds between instances of this table in the output transport stream.

Type: integer
Required: False
Minimum: 32
Maximum: 8182

patInterval
The number of milliseconds between instances of this table in the output transport stream.

Type: integer
Required: False
Minimum: 0
Maximum: 1000

programNumber
The value of the program number field in the Program Map Table.

Type: integer
Required: False
Minimum: 0
Maximum: 65535

timedMetadataPid
Packet Identifier (PID) of the timed metadata stream in the transport stream.

Type: integer
Required: False
Minimum: 32
Maximum: 8182

timedMetadata
Type: string
Required: False

scte35Source
Type: string
Required: False

MovClapAtom (enum)
When enabled, include 'clap' atom if appropriate for the video output settings.
MovCslgAtom (enum)

When enabled, file composition times will start at zero, composition times in the 'ctts' (composition time to sample) box for B-frames will be negative, and a 'cslg' (composition shift least greatest) box will be included per 14496-1 amendment 1. This improves compatibility with Apple players and tools.

MovMpeg2FourCCControl (enum)

When set to XDCAM, writes MPEG2 video streams into the QuickTime file using XDCAM fourcc codes. This increases compatibility with Apple editors and players, but may decrease compatibility with other players. Only applicable when the video codec is MPEG2.

MovPaddingControl (enum)

If set to OMNEON, inserts Omneon-compatible padding

MovReference (enum)

A value of 'external' creates separate media files and the wrapper file (.mov) contains references to these media files. A value of 'self_contained' creates only a wrapper (.mov) file and this file contains all of the media.

MovSettings

reference

Type: string
Required: False

paddingControl

Type: string
Required: False

mpeg2FourCCControl

Type: string
Required: False
cslgAtom
  Type: string
  Required: False

clapAtom
  Type: string
  Required: False

Mp2Settings

channels
Set Channels to specify the number of channels in this output audio track. Choosing Mono in the console will give you 1 output channel; choosing Stereo will give you 2. In the API, valid values are 1 and 2.
  Type: integer
  Required: False
  Minimum: 1
  Maximum: 2

bitrate
Average bitrate in bits/second.
  Type: integer
  Required: False
  Minimum: 32000
  Maximum: 384000

sampleRate
Sample rate in hz.
  Type: integer
  Required: False
  Minimum: 32000
  Maximum: 48000

Mp4CslgAtom (enum)
When enabled, file composition times will start at zero, composition times in the 'ctts' (composition time to sample) box for B-frames will be negative, and a 'cslg' (composition shift least greatest) box will be included per 14496-1 amendment 1. This improves compatibility with Apple players and tools.
  INCLUDE
  EXCLUDE

Mp4FreeSpaceBox (enum)
Inserts a free-space box immediately after the moov box.
**Mp4MoovPlacement (enum)**

If set to PROGRESSIVE_DOWNLOAD, the MOOV atom is relocated to the beginning of the archive as required for progressive downloading. Otherwise it is placed normally at the end.

- PROGRESSIVE_DOWNLOAD
- NORMAL

**Mpeg2AdaptiveQuantization (enum)**

Adaptive quantization. Allows intra-frame quantizers to vary to improve visual quality.

- OFF
- LOW
- MEDIUM
- HIGH

**Mpeg2CodecLevel (enum)**

Use Level (Mpeg2CodecLevel) to set the MPEG-2 level for the video output.

- AUTO
- LOW
- MAIN
- HIGH1440
**Mpeg2CodecProfile (enum)**

Use Profile (Mpeg2CodecProfile) to set the MPEG-2 profile for the video output.

- **MAIN**
- **PROFILE_422**

**Mpeg2FramerateControl (enum)**

If you are using the console, use the Framerate setting to specify the framerate for this output. If you want to keep the same framerate as the input video, choose Follow source. If you want to do framerate conversion, choose a framerate from the dropdown list or choose Custom. The framerates shown in the dropdown list are decimal approximations of fractions. If you choose Custom, specify your framerate as a fraction. If you are creating your transcoding job specification as a JSON file without the console, use FramerateControl to specify which value the service uses for the framerate for this output. Choose INITIALIZE_FROM_SOURCE if you want the service to use the framerate from the input. Choose SPECIFIED if you want the service to use the framerate you specify in the settings FramerateNumerator and FramerateDenominator.

- **INITIALIZE_FROM_SOURCE**
- **SPECIFIED**

**Mpeg2FramerateConversionAlgorithm (enum)**

When set to INTERPOLATE, produces smoother motion during framerate conversion.

- **DUPLICATE_DROP**
- **INTERPOLATE**

**Mpeg2GopSizeUnits (enum)**

Indicates if the GOP Size in MPEG2 is specified in frames or seconds. If seconds the system will convert the GOP Size into a frame count at run time.

- **FRAMES**
- **SECONDS**

**Mpeg2InterlaceMode (enum)**

Use Interlace mode (InterlaceMode) to choose the scan line type for the output. Top Field First (TOP_FIELD) and Bottom Field First (BOTTOM_FIELD) produce interlaced output with the entire output having the same field polarity (top or bottom first). Follow, Default Top (FOLLOW_TOP_FIELD) and Follow, Default Bottom (FOLLOW_BOTTOM_FIELD) use the same field polarity as the source. Therefore, behavior depends on the input scan type. - If the source is interlaced, the output will be interlaced with the same polarity as the source (it will follow the source). The output could therefore be a mix of "top field first" and "bottom field first". - If the source is progressive, the output will be interlaced with "top field first" or "bottom field first" polarity, depending on which of the Follow options you chose.

- **PROGRESSIVE**
- **TOP_FIELD**
- **BOTTOM_FIELD**
- **FOLLOW_TOP_FIELD**
- **FOLLOW_BOTTOM_FIELD**
**Mpeg2IntraDcPrecision (enum)**

Use Intra DC precision (Mpeg2IntraDcPrecision) to set quantization precision for intra-block DC coefficients. If you choose the value auto, the service will automatically select the precision based on the per-frame compression ratio.

- AUTO
- INTRA_DC_PRECISION_8
- INTRA_DC_PRECISION_9
- INTRA_DC_PRECISION_10
- INTRA_DC_PRECISION_11

**Mpeg2ParControl (enum)**

Using the API, enable ParFollowSource if you want the service to use the pixel aspect ratio from the input. Using the console, do this by choosing Follow source for Pixel aspect ratio.

- INITIALIZE_FROM_SOURCE
- SPECIFIED

**Mpeg2QualityTuningLevel (enum)**

Use Quality tuning level (Mpeg2QualityTuningLevel) to specify whether to use single-pass or multipass video encoding.

- SINGLE_PASS
- MULTI_PASS

**Mpeg2RateControlMode (enum)**

Use Rate control mode (Mpeg2RateControlMode) to specify whether the bitrate is variable (vbr) or constant (cbr).

- VBR
- CBR

**Mpeg2SceneChangeDetect (enum)**

Scene change detection (inserts I-frames on scene changes).

- DISABLED
- ENABLED

**Mpeg2Settings**

**minIInterval**

Enforces separation between repeated (cadence) I-frames and I-frames inserted by Scene Change Detection. If a scene change I-frame is within I-interval frames of a cadence I-frame, the GOP is shrunk and/or stretched to the scene change I-frame. GOP stretch requires enabling lookahead as well as setting I-interval. The normal cadence resumes for the next GOP. This setting is only used when Scene Change Detect is enabled. Note: Maximum GOP stretch = GOP size + Min-I-interval - 1

- **Type**: integer
- **Required**: False
Minimum: 0
Maximum: 30

parNumerator

Pixel Aspect Ratio numerator.

Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

gopSizeUnits

Type: string
Required: False

hrdBufferSize

Size of buffer (HRD buffer model) in bits. For example, enter five megabits as 5000000.

Type: integer
Required: False
Minimum: 0
Maximum: 47185920

qualityTuningLevel

Type: string
Required: False

maxBitrate

Maximum bitrate in bits/second. For example, enter five megabits per second as 5000000.

Type: integer
Required: False
Minimum: 1000
Maximum: 300000000

bitrate

Average bitrate in bits/second. Required for VBR and CBR. For MS Smooth outputs, bitrates must be unique when rounded down to the nearest multiple of 1000.

Type: integer
Required: False
Minimum: 1000
Maximum: 288000000

spatialAdaptiveQuantization

Type: string
Required: False
slowPal
  Type: string
  Required: False

intraDcPrecision
  Type: string
  Required: False

codecProfile
  Type: string
  Required: False

softness
Softness. Selects quantizer matrix, larger values reduce high-frequency content in the encoded image.
  Type: integer
  Required: False
  Minimum: 0
  Maximum: 128

framerateControl
  Type: string
  Required: False

telecine
  Type: string
  Required: False

codecLevel
  Type: string
  Required: False

framerateConversionAlgorithm
  Type: string
  Required: False

temporalAdaptiveQuantization
  Type: string
  Required: False

hrdBufferInitialFillPercentage
Percentage of the buffer that should initially be filled (HRD buffer model).
framerateNumerator

Framerate numerator - framerate is a fraction, e.g. 24000 / 1001 = 23.976 fps.

Type: integer
Required: False
Minimum: 24
Maximum: 60000

gopClosedCadence

Frequency of closed GOPs. In streaming applications, it is recommended that this be set to 1 so a decoder joining mid-stream will receive an IDR frame as quickly as possible. Setting this value to 0 will break output segmenting.

Type: integer
Required: False
Minimum: 0
Maximum: 2147483647

numberBFramesBetweenReferenceFrames

Number of B-frames between reference frames.

Type: integer
Required: False
Minimum: 0
Maximum: 7

framerateDenominator

Framerate denominator.

Type: integer
Required: False
Minimum: 1
Maximum: 1001

adaptiveQuantization

Type: string
Required: False

interlaceMode

Type: string
Required: False
**gopSize**

GOP Length (keyframe interval) in frames or seconds. Must be greater than zero.

- **Type:** number
- **Required:** False
- **Format:** float
- **Minimum:** 0.0

**syntax**

- **Type:** string
- **Required:** False

**parDenominator**

Pixel Aspect Ratio denominator.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 2147483647

**sceneChangeDetect**

- **Type:** string
- **Required:** False

**parControl**

- **Type:** string
- **Required:** False

**rateControlMode**

- **Type:** string
- **Required:** False

**Mpeg2SlowPal (enum)**

Enables Slow PAL rate conversion. 23.976fps and 24fps input is relabeled as 25fps, and audio is sped up correspondingly.

- DISABLED
- ENABLED

**Mpeg2SpatialAdaptiveQuantization (enum)**

Adjust quantization within each frame based on spatial variation of content complexity.

- DISABLED
- ENABLED
**Mpeg2Syntax (enum)**

Produces a Type D-10 compatible bitstream (SMPTE 356M-2001).

- DEFAULT
- D_10

**Mpeg2Telecine (enum)**

Only use Telecine (Mpeg2Telecine) when you set Framerate (Framerate) to 29.970. Set Telecine (Mpeg2Telecine) to Hard (hard) to produce a 29.97i output from a 23.976 input. Set it to Soft (soft) to produce 23.976 output and leave conversion to the player.

- NONE
- SOFT
- HARD

**Mpeg2TemporalAdaptiveQuantization (enum)**

Adjust quantization within each frame based on temporal variation of content complexity.

- DISABLED
- ENABLED

**NoiseReducer**

**filter**

Type: string
Required: True

**filterSettings**

Type: NoiseReducerFilterSettings (p. 740)
Required: False

**spatialFilterSettings**

Type: NoiseReducerSpatialFilterSettings (p. 740)
Required: False

**NoiseReducerFilter (enum)**

Use Noise reducer filter (NoiseReducerFilter) to select one of the following spatial image filtering functions. To use this setting, you must also enable Noise reducer (NoiseReducer).
- Bilateral is an edge preserving noise reduction filter.
- Mean (softest), Gaussian, Lanczos, and Sharpen (sharpest) are convolution filters.
- Conserve is a min/max noise reduction filter.
- Spatial is a frequency-domain filter based on JND principles.

- BILATERAL
- MEAN
- GAUSSIAN
- LANCZOS
SHARPEN
CONSERVE
SPATIAL

NoiseReducerFilterSettings

strength
Relative strength of noise reducing filter. Higher values produce stronger filtering.

Type: integer
Required: False
Minimum: 0
Maximum: 3

NoiseReducerSpatialFilterSettings

strength
Relative strength of noise reducing filter. Higher values produce stronger filtering.

Type: integer
Required: False
Minimum: 0
Maximum: 16

postFilterSharpenStrength
Specify strength of post noise reduction sharpening filter, with 0 disabling the filter and 3 enabling it at maximum strength.

Type: integer
Required: False
Minimum: 0
Maximum: 3

speed
The speed of the filter, from -2 (lower speed) to 3 (higher speed), with 0 being the nominal value.

Type: integer
Required: False
Minimum: -2
Maximum: 3

Order (enum)
When you request lists of resources, you can optionally specify whether they are sorted in ASCENDING or DESCENDING order. Default varies by resource.

ASCENDING
DESCENDING
OutputChannelMapping

inputChannels
List of input channels

Type: Array of type integer
Required: True

OutputSdt (enum)
Selects method of inserting SDT information into output stream. "Follow input SDT" copies SDT information from input stream to output stream. "Follow input SDT if present" copies SDT information from input stream to output stream if SDT information is present in the input, otherwise it will fall back on the user-defined values. Enter "SDT Manually" means user will enter the SDT information. "No SDT" means output stream will not contain SDT information.

SDT_FOLLOW
SDT_FOLLOW_IF_PRESENT
SDT_MANUAL
SDT_NONE

Preset

createdAt
The timestamp in epoch seconds for preset creation.

Type: string
Required: False
Format: date-time

lastUpdated
The timestamp in epoch seconds when the preset was last updated.

Type: string
Required: False
Format: date-time

settings

Type: PresetSettings (p. 742)
Required: True

name
A name you create for each preset. Each name must be unique within your account.

Type: string
Required: True

description
An optional description you create for each preset.
Properties

Type: string
Required: False

arn
An identifier for this resource that is unique within all of AWS.

Type: string
Required: False

category
An optional category you create to organize your presets.

Type: string
Required: False

type
A preset can be of two types: system or custom. System or built-in preset can't be modified or deleted by the user.

Type: string
Required: False

PresetListBy (enum)
Optional. When you request a list of presets, you can choose to list them alphabetically by NAME or chronologically by CREATION_DATE. If you don't specify, the service will list them by name.

NAME
CREATION_DATE
SYSTEM

PresetSettings

videoDescription
(VideoDescription) contains a group of video encoding settings. The specific video settings depend on the video codec you choose when you specify a value for Video codec (codec). Include one instance of (VideoDescription) per output.

Type: VideoDescription (p. 750)
Required: False

audioDescriptions
(AudioDescriptions) contains groups of audio encoding settings organized by audio codec. Include one instance of (AudioDescriptions) per output. (AudioDescriptions) can contain multiple groups of encoding settings.

Type: Array of type AudioDescription (p. 669)
Required: False

containerSettings

Type: ContainerSettings (p. 679)
Required: False

captionDescriptions

Caption settings for this preset. There can be multiple caption settings in a single output.

Type: Array of type CaptionDescriptionPreset (p. 676)
Required: False

ProresCodecProfile (enum)

Use Profile (ProResCodecProfile) to specify the type of Apple ProRes codec to use for this output.

APPLE_PRORES_422
APPLE_PRORES_422_HQ
APPLE_PRORES_422_LT
APPLE_PRORES_422_PROXY

ProresFramerateControl (enum)

If you are using the console, use the Framerate setting to specify the framerate for this output. If you want to keep the same framerate as the input video, choose Follow source. If you want to do framerate conversion, choose a framerate from the dropdown list or choose Custom. The framerates shown in the dropdown list are decimal approximations of fractions. If you choose Custom, specify your framerate as a fraction. If you are creating your transcoding job specification as a JSON file without the console, use FramerateControl to specify which value the service uses for the framerate for this output. Choose INITIALIZE_FROM_SOURCE if you want the service to use the framerate from the input. Choose SPECIFIED if you want the service to use the framerate you specify in the settings FramerateNumerator and FramerateDenominator.

INITIALIZE_FROM_SOURCE
SPECIFIED

ProresFramerateConversionAlgorithm (enum)

When set to INTERPOLATE, produces smoother motion during framerate conversion.

DUPLICATE_DROP
INTERPOLATE

ProresInterlaceMode (enum)

Use Interlace mode (InterlaceMode) to choose the scan line type for the output. * Top Field First (TOP_FIELD) and Bottom Field First (BOTTOM_FIELD) produce interlaced output with the entire output having the same field polarity (top or bottom first). * Follow, Default Top (FOLLOW_TOP_FIELD) and Follow, Default Bottom (FOLLOW_BOTTOM_FIELD) use the same field polarity as the source. Therefore, behavior depends on the input scan type. - If the source is interleaved, the output will be interleaved with the same polarity as the source (it will follow the source). The output could therefore be a mix of "top field first" and "bottom field first". - If the source is progressive, the output will be interlaced with "top field first" or "bottom field first" polarity, depending on which of the Follow options you chose.
PROGRESSIVE
TOP_FIELD
BOTTOM_FIELD
FOLLOW_TOP_FIELD
FOLLOW_BOTTOM_FIELD

**ProresParControl (enum)**

Use (ProresParControl) to specify how the service determines the pixel aspect ratio. Set to Follow source (INITIALIZE_FROM_SOURCE) to use the pixel aspect ratio from the input. To specify a different pixel aspect ratio: Using the console, choose it from the dropdown menu. Using the API, set ProresParControl to (SPECIFIED) and provide for (ParNumerator) and (ParDenominator).

- INITIALIZE_FROM_SOURCE
- SPECIFIED

**ProresSettings**

**framerateDenominator**

Framerate denominator.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 2147483647

**slowPal**

- **Type:** string
- **Required:** False

**framerateControl**

- **Type:** string
- **Required:** False

**telecine**

- **Type:** string
- **Required:** False

**framerateConversionAlgorithm**

- **Type:** string
- **Required:** False

**interlaceMode**

- **Type:** string
- **Required:** False
**parNumerator**

Pixel Aspect Ratio numerator.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 2147483647

**codecProfile**

- **Type:** string
- **Required:** False

**parDenominator**

Pixel Aspect Ratio denominator.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 2147483647

**parControl**

- **Type:** string
- **Required:** False

**framerateNumerator**

When you use the API for transcode jobs that use framerate conversion, specify the framerate as a fraction. For example, 24000 / 1001 = 23.976 fps. Use FramerateNumerator to specify the numerator of this fraction. In this example, use 24000 for the value of FramerateNumerator.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 2147483647

**ProresSlowPal (enum)**

Enables Slow PAL rate conversion. 23.976fps and 24fps input is relabeled as 25fps, and audio is sped up correspondingly.

- **DISABLED**
- **ENABLED**

**ProresTelecine (enum)**

Only use Telecine (ProresTelecine) when you set Framerate (Framerate) to 29.970. Set Telecine (ProresTelecine) to Hard (hard) to produce a 29.97i output from a 23.976 input. Set it to Soft (soft) to produce 23.976 output and leave conversion to the player.

- **NONE**
HARD

**Rectangle**

**width**
Width of rectangle in pixels.

*Type:* integer  
*Required:* True  
*Minimum:* -2147483648  
*Maximum:* 2147483647

**x**
The distance, in pixels, between the rectangle and the left edge of the video frame.

*Type:* integer  
*Required:* True  
*Minimum:* -2147483648  
*Maximum:* 2147483647

**y**
The distance, in pixels, between the rectangle and the top edge of the video frame.

*Type:* integer  
*Required:* True  
*Minimum:* -2147483648  
*Maximum:* 2147483647

**height**
Height of rectangle in pixels.

*Type:* integer  
*Required:* True  
*Minimum:* -2147483648  
*Maximum:* 2147483647

**RemixSettings**

**channelsOut**
Specify the number of channels in this output after remixing. Valid values: 1, 2, 4, 6, 8

*Type:* integer  
*Required:* True  
*Minimum:* 1  
*Maximum:* 8

**channelMapping**

*Type:* [ChannelMapping](p. 678)
Required: True

channelsIn

Specify the number of audio channels from your input that you want to use in your output. With remixing, you might combine or split the data in these channels, so the number of channels in your final output might be different.

Type: integer
Required: True
Minimum: 1
Maximum: 16

RespondToAfd (enum)

Use Respond to AFD (RespondToAfd) to specify how the service changes the video itself in response to AFD values in the input. * Choose Respond to clip the input video frame according to the AFD value, input display aspect ratio, and output display aspect ratio. * Choose Passthrough to include the input AFD values. Do not choose this when AfdSignaling is set to (NONE). A preferred implementation of this workflow is to set RespondToAfd to (NONE) and set AfdSignaling to (AUTO). * Choose None to remove all input AFD values from this output.

NONE
RESPOND
PASSTHROUGH

ScalingBehavior (enum)

Applies only if your input aspect ratio is different from your output aspect ratio. Enable Stretch to output (StretchToOutput) to have the service stretch your video image to fit. Leave this setting disabled to allow the service to letterbox your video instead. This setting overrides any positioning value you specify elsewhere in the job.

DEFAULT
STRETCH_TO_OUTPUT

SccDestinationFramerate (enum)

Set Framerate (SccDestinationFramerate) to make sure that the captions and the video are synchronized in the output. Specify a framerate that matches the framerate of the associated video. If the video framerate is 29.97, choose 29.97 dropframe (FRAMERATE_29_97_DROPFRAME) only if the video has video_insertion=true and drop_frame_timecode=true; otherwise, choose 29.97 non-dropframe (FRAMERATE_29_97_NON_DROPFRAME).

FRAMERATE_23_97
FRAMERATE_24
FRAMERATE_29_97_DROPFRAME
FRAMERATE_29_97_NON_DROPFRAME

SccDestinationSettings

framerate

Type: string
**Properties**

**Required**: False

### TeletextDestinationSettings

**pageNumber**

Set `pageNumber` to the Teletext page number for the destination captions for this output. This value must be a three-digit hexadecimal string; strings ending in -FF are invalid. If you are passing through the entire set of Teletext data, do not use this field.

*Type*: string  
*Required*: False  
*Pattern*: `^[1-8][0-9a-fA-F][0-9a-eA-E]$`

### TimecodeBurnIn

**prefix**

Use `Prefix` (Prefix) to place ASCII characters before any burned-in timecode. For example, a prefix of "EZ-" will result in the timecode "EZ-00:00:00:00". Provide either the characters themselves or the ASCII code equivalents. The supported range of characters is 0x20 through 0x7e. This includes letters, numbers, and all special characters represented on a standard English keyboard.

*Type*: string  
*Required*: False  
*Pattern*: `^[ -~]+$`

**fontSize**

Use `Font Size` (FontSize) to set the font size of any burned-in timecode. Valid values are 10, 16, 32, 48.

*Type*: integer  
*Required*: False  
*Minimum*: 10  
*Maximum*: 48

**position**

*Type*: string  
*Required*: False

### TimecodeBurninPosition (enum)

Use `Position` (Position) under `Timecode Burnin` (TimecodeBurnIn) to specify the location the burned-in timecode on output video.

- `TOP_CENTER`
- `TOP_LEFT`
- `TOP_RIGHT`
- `MIDDLE_LEFT`
- `MIDDLE_CENTER`
- `MIDDLE_RIGHT`
TimedMetadata (enum)

Applies only to HLS outputs. Use this setting to specify whether the service inserts the ID3 timed metadata from the input in this output.

- PASSTHROUGH
- NONE

TtmlDestinationSettings

stylePassthrough

Type: string

Required: False

TtmlStylePassthrough (enum)

Pass through style and position information from a TTML-like input source (TTML, SMPTE-TT, CFF-TT) to the CFF-TT output or TTML output.

- ENABLED
- DISABLED

Type (enum)

- SYSTEM
- CUSTOM

VideoCodec (enum)

Type of video codec

- FRAME_CAPTURE
- H_264
- H_265
- MPEG2
- PRORES

VideoCodecSettings

h265Settings

Type: H265Settings

Required: False

codec

Type: string
Required: True

proresSettings

Type: ProresSettings (p. 744)
Required: False

mpeg2Settings

Type: Mpeg2Settings (p. 734)
Required: False

frameCaptureSettings

Type: FrameCaptureSettings (p. 692)
Required: False

h264Settings

Type: H264Settings (p. 696)
Required: False

VideoDescription

fixedAfd
Apply only if you set AFD Signaling(AfdSignaling) to Fixed (FIXED). Use Fixed (FixedAfd) to specify a four-bit AFD value which the service will write on all frames of this video output.

Type: integer
Required: False
Minimum: 0
Maximum: 15

scalingBehavior

Type: string
Required: False

respondToAfd

Type: string
Required: False

codecSettings

Type: VideoCodecSettings (p. 749)
Required: True

afdSignaling

Type: string
Required: False

colorMetadata

Type: string
Required: False

timecodeInsertion

Type: string
Required: False

width

Use Width (Width) to define the video resolution width, in pixels, for this output. If you don't provide a value here, the service will use the input width.

Type: integer
Required: False
Minimum: 32
Maximum: 4096

videoPreprocessors

Find additional transcoding features under Preprocessors (VideoPreprocessors). Enable the features at each output individually. These features are disabled by default.

Type: VideoPreprocessor (p. 752)
Required: False

antiAlias

Type: string
Required: False

position

Use Position (Position) to point to a rectangle object to define your position. This setting overrides any other aspect ratio.

Type: Rectangle (p. 746)
Required: False

sharpness

Use Sharpness (Sharpness)setting to specify the strength of anti-aliasing. This setting changes the width of the anti-alias filter kernel used for scaling. Sharpness only applies if your output resolution is different from your input resolution, and if you set Anti-alias (AntiAlias) to ENABLED. 0 is the softest setting, 100 the sharpest, and 50 recommended for most content.

Type: integer
Required: False
Minimum: 0  
Maximum: 100

crop

Applies only if your input aspect ratio is different from your output aspect ratio. Use Input cropping rectangle (Crop) to specify the video area the service will include in the output. This will crop the input source, causing video pixels to be removed on encode. Do not use this setting if you have enabled Stretch to output (stretchToOutput) in your output settings.

Type: Rectangle (p. 746)  
Required: False

dropFrameTimecode

Type: string  
Required: False

height

Use the Height (Height) setting to define the video resolution height for this output. Specify in pixels. If you don’t provide a value here, the service will use the input height.

Type: integer  
Required: False  
Minimum: 32  
Maximum: 2160

VideoPreprocessor

noiseReducer

Enable the Noise reducer (NoiseReducer) feature to remove noise from your video output if necessary. Enable or disable this feature for each output individually. This setting is disabled by default.

Type: NoiseReducer (p. 739)  
Required: False

timecodeBurnin

Timecode burn-in (TimecodeBurnIn)—Burns the output timecode and specified prefix into the output.

Type: TimecodeBurnin (p. 748)  
Required: False

colorCorrector

Enable the Color corrector (ColorCorrector) feature if necessary. Enable or disable this feature for each output individually. This setting is disabled by default.

Type: ColorCorrector (p. 678)  
Required: False
**deinterlacer**

Use Deinterlacer (Deinterlacer) to produce smoother motion and a clearer picture.

*Type: Deinterlacer (p. 681)*
*Required: False*

**imageInserter**

Enable the Image inserter (ImageInserter) feature to include a graphic overlay on your video. Enable or disable this feature for each output individually. This setting is disabled by default.

*Type: ImageInserter (p. 713)*
*Required: False*

**VideoTimecodeInsertion (enum)**

Applies only to H.264, H.265, MPEG2, and ProRes outputs. Only enable Timecode insertion when the input framerate is identical to the output framerate. To include timecodes in this output, set Timecode insertion (VideoTimecodeInsertion) to PIC_TIMING_SEI. To leave them out, set it to DISABLED. Default is DISABLED. When the service inserts timecodes in an output, by default, it uses any embedded timecodes from the input. If none are present, the service will set the timecode for the first output frame to zero. To change this default behavior, adjust the settings under Timecode configuration (TimecodeConfig). In the console, these settings are located under Job > Job settings > Timecode configuration. Note - Timecode source under input settings (InputTimecodeSource) does not affect the timecodes that are inserted in the output. Source under Job settings > Timecode configuration (TimecodeSource) does.

*DISABLED*
*PIC_TIMING_SEI*

**WavFormat (enum)**

The service defaults to using RIFF for WAV outputs. If your output audio is likely to exceed 4 GB in file size, or if you otherwise need the extended support of the RF64 format, set your output WAV file format to RF64.

*RIFF*
*RF64*

**WavSettings**

**channels**

Set Channels to specify the number of channels in this output audio track. With WAV, valid values 1, 2, 4, and 8. In the console, these values are Mono, Stereo, 4-Channel, and 8-Channel, respectively.

*Type: integer*
*Required: False*
*Minimum: 1*
*Maximum: 8*

**bitDepth**

Specify Bit depth (BitDepth), in bits per sample, to choose the encoding quality for this audio track.
Presets name

URI

/2017-08-29/presets/ name

HTTP Methods

GET

Operation ID: GetPreset

Retrieve the JSON for a specific preset.

Path Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>True</td>
<td></td>
</tr>
</tbody>
</table>

Responses

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>GetPresetResponse (p. 763)</td>
<td>200 response</td>
</tr>
<tr>
<td>400</td>
<td>ExceptionBody (p. 778)</td>
<td>BadRequestException</td>
</tr>
<tr>
<td>500</td>
<td>ExceptionBody (p. 778)</td>
<td>InternalServiceException</td>
</tr>
<tr>
<td>403</td>
<td>ExceptionBody (p. 778)</td>
<td>AccessDeniedException</td>
</tr>
<tr>
<td>404</td>
<td>ExceptionBody (p. 778)</td>
<td>ResourceNotFoundException</td>
</tr>
</tbody>
</table>
PUT

Operation ID: UpdatePreset

Modify one of your existing presets.

Path Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>True</td>
<td></td>
</tr>
</tbody>
</table>

Responses

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>UpdatePresetResponse (p. 778)</td>
<td>200 response</td>
</tr>
<tr>
<td>400</td>
<td>ExceptionBody (p. 778)</td>
<td>BadRequestException</td>
</tr>
<tr>
<td>500</td>
<td>ExceptionBody (p. 778)</td>
<td>InternalServiceException</td>
</tr>
<tr>
<td>403</td>
<td>ExceptionBody (p. 778)</td>
<td>AccessDeniedException</td>
</tr>
<tr>
<td>404</td>
<td>ExceptionBody (p. 778)</td>
<td>ResourceNotFoundException</td>
</tr>
<tr>
<td>429</td>
<td>ExceptionBody (p. 778)</td>
<td>LimitExceededException</td>
</tr>
<tr>
<td>409</td>
<td>ExceptionBody (p. 778)</td>
<td>ResourceInUseException</td>
</tr>
</tbody>
</table>

DELETE

Operation ID: DeletePreset

Permanently delete a preset you have created.

Path Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>True</td>
<td></td>
</tr>
</tbody>
</table>

Responses

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>ExceptionBody (p. 778)</td>
<td>BadRequestException</td>
</tr>
<tr>
<td>202</td>
<td>DeletePresetResponse (p. 778)</td>
<td>202 response</td>
</tr>
<tr>
<td>Status Code</td>
<td>Response Model</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>500</td>
<td>ExceptionBody (p. 778)</td>
<td>InternalServiceException</td>
</tr>
<tr>
<td>403</td>
<td>ExceptionBody (p. 778)</td>
<td>AccessDeniedException</td>
</tr>
<tr>
<td>404</td>
<td>ExceptionBody (p. 778)</td>
<td>ResourceNotFoundException</td>
</tr>
<tr>
<td>429</td>
<td>ExceptionBody (p. 778)</td>
<td>LimitExceededException</td>
</tr>
<tr>
<td>409</td>
<td>ExceptionBody (p. 778)</td>
<td>ResourceInUseException</td>
</tr>
</tbody>
</table>

## Schemas

### Request Bodies

#### Example GET

```json
{
    "name (p. 808)": "string"
}
```

#### Example PUT

```json
{
    "settings (p. 863)": {
        "videoDescription (p. 856)": {
            "fixedAfd (p. 864)": integer,
            "scalingBehavior (p. 864)": enum,
            "respondToAfd (p. 864)": enum,
            "codecSettings (p. 864)": {
                "h265Settings (p. 863)": {
                    "slices (p. 819)": integer,
                    "minIInterval (p. 820)": integer,
                    "parNumerator (p. 820)": integer,
                    "flickerAdaptiveQuantization (p. 820)": enum,
                    "gopSizeUnits (p. 820)": enum,
                    "hrdBufferSize (p. 820)": integer,
                    "qualityTuningLevel (p. 820)": enum,
                    "maxBitrate (p. 820)": integer,
                    "bitrate (p. 821)": integer,
                    "spatialAdaptiveQuantization (p. 821)": enum,
                    "sampleAdaptiveOffsetFilterMode (p. 821)": enum,
                    "temporalIds (p. 821)": enum,
                    "slowPal (p. 821)": enum,
                    "tiles (p. 821)": enum,
                    "writeMp4PackagingType (p. 821)": enum,
                    "codecProfile (p. 821)": enum,
                    "alternateTransferFunctionSei (p. 821)": enum,
                    "unregisteredSeiTimecode (p. 822)": enum,
                    "framerateControl (p. 822)": enum,
                    "telecine (p. 822)": enum,
                    "codecLevel (p. 822)": enum,
                    "framerateConversionAlgorithm (p. 822)": enum,
                    "numberReferenceFrames (p. 822)": integer,
                    "temporalAdaptiveQuantization (p. 822)": enum,
                    "hrdBufferInitialFillPercentage (p. 822)": integer,
                    "framerateNumerator (p. 822)": integer,
                    "gopClosedCadence (p. 823)": integer,
```
"numberBFramesBetweenReferenceFrames (p. 823)": integer,
"framerateDenominator (p. 823)": integer,
"adaptiveQuantization (p. 823)": enum,
"interlaceMode (p. 823)": enum,
"gopSize (p. 823)": number,
"gopBReference (p. 824)": enum,
"parDenominator (p. 824)": integer,
"sceneChangeDetect (p. 824)": enum,
"parControl (p. 824)": enum,
"rateControlMode (p. 824)": enum
},
"codec (p. 864)": enum,
"proresSettings (p. 864)": {
  "framerateDenominator (p. 857)": integer,
  "slowPal (p. 858)": enum,
  "framerateControl (p. 858)": enum,
  "telecine (p. 858)": enum,
  "framerateConversionAlgorithm (p. 858)": enum,
  "interlaceMode (p. 858)": enum,
  "parDenominator (p. 858)": integer,
  "codecProfile (p. 858)": enum,
  "parDenominator (p. 858)": integer,
  "parControl (p. 858)": enum,
  "framerateNumerator (p. 859)": integer
},
"mpeg2Settings (p. 864)": {
  "minIInterval (p. 848)": integer,
  "parNumerator (p. 848)": integer,
  "gopSizeUnits (p. 848)": enum,
  "hrdBufferSize (p. 849)": integer,
  "qualityTuningLevel (p. 849)": enum,
  "maxBitrate (p. 849)": integer,
  "bitrate (p. 849)": integer,
  "spatialAdaptiveQuantization (p. 849)": enum,
  "slowPal (p. 849)": enum,
  "intraDcPrecision (p. 849)": enum,
  "codecProfile (p. 849)": enum,
  "softness (p. 850)": integer,
  "framerateControl (p. 850)": enum,
  "telecine (p. 850)": enum,
  "codecLevel (p. 850)": enum,
  "framerateConversionAlgorithm (p. 850)": enum,
  "temporalAdaptiveQuantization (p. 850)": enum,
  "hrdBufferInitialFillPercentage (p. 850)": integer,
  "framerateNumerator (p. 850)": integer,
  "gopClosedCadence (p. 851)": integer,
  "numberBFramesBetweenReferenceFrames (p. 851)": integer,
  "adaptiveQuantization (p. 851)": integer,
  "interlaceMode (p. 851)": enum,
  "gopSize (p. 851)": number,
  "syntax (p. 851)": enum,
  "parDenominator (p. 852)": integer,
  "sceneChangeDetect (p. 852)": enum,
  "parControl (p. 852)": enum,
  "rateControlMode (p. 852)": enum
},
"frameCaptureSettings (p. 864)": {
  "framerateDenominator (p. 807)": integer,
  "maxCaptures (p. 807)": integer,
  "framerateNumerator (p. 807)": integer,
  "quality (p. 807)": integer
},
"h264Settings (p. 864)": {
  "slices (p. 811)": integer,
  "minIInterval (p. 811)": integer,
"parNumerator (p. 811)" : integer,
"flickerAdaptiveQuantization (p. 811)" : enum,
"gopSizeUnits (p. 811)" : enum,
"hrdBufferSize (p. 811)" : integer,
"qualityTuningLevel (p. 812)" : enum,
"maxBitrate (p. 812)" : integer,
"bitrate (p. 812)" : integer,
"spatialAdaptiveQuantization (p. 812)" : enum,
"slowPal (p. 812)" : enum,
"codecProfile (p. 812)" : enum,
"unregisteredSeiTimecode (p. 812)" : enum,
"softness (p. 812)" : integer,
"framerateControl (p. 813)" : enum,
"telecine (p. 813)" : enum,
"codecLevel (p. 813)" : enum,
"framerateConversionAlgorithm (p. 813)" : enum,
"numberReferenceFrames (p. 813)" : integer,
"temporalAdaptiveQuantization (p. 813)" : enum,
"repeatPps (p. 813)" : enum,
"hrdBufferInitialFillPercentage (p. 813)" : integer,
"framerateDenominator (p. 814)" : integer,
"gopClosedCadence (p. 814)" : integer,
"numberBFramesBetweenReferenceFrames (p. 814)" : integer,
"framerateNumerator (p. 814)" : integer,
"entropyEncoding (p. 814)" : enum,
"fieldEncoding (p. 814)" : enum,
"adaptiveQuantization (p. 815)" : enum,
"interlaceMode (p. 815)" : enum,
"gopSize (p. 815)" : number,
"gopBReference (p. 815)" : enum,
"syntax (p. 815)" : enum,
"parDenominator (p. 815)" : integer,
"sceneChangeDetect (p. 815)" : enum,
"rateControlMode (p. 815)" : enum
},
"afdSignaling (p. 865)" : enum,
"colorMetadata (p. 865)" : enum,
"timecodeInsertion (p. 865)" : enum,
"width (p. 865)" : integer,
"videoPreprocessors (p. 865)" : {
  "noiseReducer (p. 866)" : {
    "filter (p. 853)" : enum,
    "filterSettings (p. 853)" : {
      "strength (p. 853)" : integer
    },
    "spatialFilterSettings (p. 853)" : {
      "strength (p. 854)" : integer,
      "postFilterSharpenStrength (p. 854)" : integer,
      "speed (p. 854)" : integer
    }
  },
  "timecodeBurnin (p. 866)" : {
    "prefix (p. 861)" : "string",
    "fontSize (p. 861)" : integer,
    "position (p. 862)" : enum
  },
  "colorCorrector (p. 866)" : {
    "saturation (p. 793)" : integer,
    "brightness (p. 793)" : integer,
    "hdr10Metadata (p. 793)" : {
      "redPrimaryY (p. 825)" : integer,
      "greenPrimaryY (p. 826)" : integer,
      "whitePointX (p. 826)" : integer,
      "maxLuminance (p. 826)" : integer,
      "maxLuminance (p. 826)" : integer
    }
  }
}
"greenPrimaryX (p. 826)": integer,
"whitePointY (p. 826)": integer,
"redPrimaryX (p. 827)": integer,
"bluePrimaryX (p. 827)": integer,
"bluePrimaryY (p. 827)": integer,
"maxFrameAverageLightLevel (p. 827)": integer,
"maxContentLightLevel (p. 827)": integer,
"minLuminance (p. 828)": integer,
"contrast (p. 793)": integer,
"hue (p. 793)": integer,
"colorSpaceConversion (p. 794)": enum,
"deinterlacer (p. 867)": {
  "mode (p. 795)": enum,
  "control (p. 795)": enum,
  "algorithm (p. 795)": enum
},
"imageInserter (p. 867)": {
  "insertableImages (p. 828)": [
    {
      "duration (p. 828)": integer,
      "fadeOut (p. 828)": integer,
      "imageY (p. 828)": integer,
      "fadeIn (p. 828)": integer,
      "imageX (p. 829)": integer,
      "width (p. 829)": integer,
      "startTime (p. 829)": "string",
      "opacity (p. 829)": integer,
      "layer (p. 829)": integer,
      "height (p. 830)": integer,
      "imageInserterInput (p. 830)": "string"
    }
  ]
},
"antiAlias (p. 865)": enum,
"position (p. 865)": {
  "width (p. 859)": integer,
  "x (p. 859)": integer,
  "y (p. 859)": integer,
  "height (p. 860)": integer
},
"sharpness (p. 865)": integer,
"crop (p. 866)": {
  "width (p. 859)": integer,
  "x (p. 859)": integer,
  "y (p. 859)": integer,
  "height (p. 860)": integer
},
"dropFrameTimecode (p. 866)": enum,
"height (p. 866)": integer
},
"audioDescriptions (p. 856)": [
  {
    "audioTypeControl (p. 784)": enum,
    "languageCodeControl (p. 784)": enum,
    "remixSettings (p. 784)": {
      "channelsOut (p. 860)": integer,
      "channelMapping (p. 860)": {
        "outputChannels (p. 793)": [
          {"inputChannels (p. 854)": [integer]
        }
      }
    }
  }
]
"channelsIn (p. 860)": integer,
"audioType (p. 784)": integer,
"audioSourceName (p. 785)": "string",
"codecSettings (p. 785)": {
    "codec (p. 783)": enum,
    "wavSettings (p. 783)": {
        "channels (p. 867)": integer,
        "bitDepth (p. 867)": integer,
        "format (p. 868)": enum,
        "sampleRate (p. 868)": integer
    },
    "aacSettings (p. 784)": {
        "vbrQuality (p. 779)": enum,
        "codecProfile (p. 779)": enum,
        "codingMode (p. 779)": enum,
        "specification (p. 779)": enum,
        "bitrate (p. 779)": integer,
        "rawFormat (p. 779)": enum,
        "rateControlMode (p. 780)": enum,
        "sampleRate (p. 780)": integer,
        "audioDescriptionBroadcasterMix (p. 780)": enum
    },
    "ac3Settings (p. 784)": {
        "dynamicRangeCompressionProfile (p. 781)": enum,
        "dialnorm (p. 781)": integer,
        "codingMode (p. 781)": enum,
        "metadataControl (p. 781)": enum,
        "bitrate (p. 782)": integer,
        "lfeFilter (p. 782)": enum,
        "bitstreamMode (p. 782)": enum,
        "sampleRate (p. 782)": integer
    },
    "aiffSettings (p. 784)": {
        "channels (p. 782)": integer,
        "bitDepth (p. 783)": integer,
        "sampleRate (p. 783)": integer
    },
    "eac3Settings (p. 784)": {
        "dialnorm (p. 803)": integer,
        "passthroughControl (p. 803)": enum,
        "metadataControl (p. 803)": enum,
        "bitrate (p. 803)": integer,
        "dynamicRangeCompressionLfe (p. 804)": enum,
        "sampleRate (p. 804)": integer,
        "surroundExMode (p. 804)": enum,
        "ltRtSurroundMixLevel (p. 804)": number,
        "dynamicRangeCompressionLine (p. 804)": enum,
        "lfeControl (p. 804)": enum,
        "codingMode (p. 804)": enum,
        "surroundMode (p. 804)": enum,
        "attenuationControl (p. 805)": enum,
        "lfeFilter (p. 805)": enum,
        "phaseControl (p. 805)": enum,
        "ltRtCenterMixLevel (p. 805)": number,
        "dcFilter (p. 805)": enum,
        "bitstreamMode (p. 805)": enum,
        "stereoDownmix (p. 805)": enum,
        "loRoSurroundMixLevel (p. 805)": number,
        "loRoCenterMixLevel (p. 806)": number
    },
    "mp2Settings (p. 784)": {
        "channels (p. 844)": integer,
        "bitrate (p. 845)": integer,
"samplingRate (p. 845)": integer
",
"languageCode (p. 785)": enum,
"streamName (p. 785)": "string",
"audioNormalizationSettings (p. 785)": {
  "targetLkfs (p. 786)": number,
  "algorithmControl (p. 787)": enum,
  "loudnessLogging (p. 787)": enum,
  "peakCalculation (p. 787)": enum,
  "correctionGateLevel (p. 787)": integer,
  "algorithm (p. 787)": enum
},
"customLanguageCode (p. 785)": "string"
",
"containerSettings (p. 856)": {
  "container (p. 794)": enum,
  "mp4Settings (p. 794)": {
    "mp4MajorBrand (p. 845)": "string",
    "moovPlacement (p. 846)": enum,
    "cslgAtom (p. 846)": enum,
    "freeSpaceBox (p. 846)": enum
  },
  "m3u8Settings (p. 794)": {
    "pmtPid (p. 841)": integer,
    "nielsenId3 (p. 841)": enum,
    "pcrPid (p. 841)": integer,
    "audioPids (p. 841)": [integer
      ]
  },
  "audioFramesPerPES (p. 841)": integer,
  "scte35Pid (p. 841)": integer,
  "transportStreamId (p. 842)": integer,
  "pcrControl (p. 842)": enum,
  "videoPid (p. 842)": integer,
  "privateMetadataPid (p. 842)": integer,
  "pmtInterval (p. 842)": integer,
  "patInterval (p. 842)": integer,
  "programNumber (p. 843)": integer,
  "timedMetadataPid (p. 843)": integer,
  "timedMetadata (p. 843)": enum,
  "scte35Source (p. 843)": enum
},
"m2tsSettings (p. 794)": {
  "dvbTeletextPid (p. 835)": integer,
  "nielsenId3 (p. 836)": enum,
  "bitrate (p. 836)": integer,
  "segmentationTime (p. 836)": number,
  "audioPids (p. 836)": [integer
    ]
},
"rateMode (p. 836)": enum,
"maxPcrInterval (p. 836)": integer,
"audioFramesPerPES (p. 836)": integer,
"ebpAudioInterval (p. 837)": enum,
"fragmentTime (p. 837)": number,
"scte35Pid (p. 837)": integer,
"pmtInterval (p. 837)": integer,
"privateMetadataPid (p. 837)": integer,
"segmentationStyle (p. 837)": enum,
"audioBufferModel (p. 837)": enum,
"programNumber (p. 838)": integer,
"dvbNitSettings (p. 838)": {
  "networkName (p. 796)": "string",
  "networkId (p. 796)": integer,
"nitInterval (p. 797)": integer,
"timedMetadataPid (p. 838)": integer,
"scte35Source (p. 838)": enum,
"pmtPid (p. 838)": integer,
"bufferModel (p. 838)": enum,
"ebpPlacement (p. 838)": enum,
"dvbSdtSettings (p. 838)": {
  "sdInterval (p. 797)": integer,
  "serviceName (p. 797)": "string",
  "serviceProviderName (p. 797)": "string",
  "outputSdt (p. 797)": enum
},
"nullPacketBitrate (p. 839)": number,
"pcrPid (p. 839)": integer,
"minEbpInterval (p. 839)": integer,
"transportStreamId (p. 839)": integer,
"videoPid (p. 839)": integer,
"pcrControl (p. 840)": enum,
"esRateInPes (p. 840)": enum,
"segmentationMarkers (p. 840)": enum,
"dvbTdtSettings (p. 840)": {
  "tdtInterval (p. 801)": integer
},
"patInterval (p. 840)": integer,
"dvbSubPids (p. 840)": [ integer ]
},
"movSettings (p. 794)": {
  "reference (p. 844)": enum,
  "paddingControl (p. 844)": enum,
  "mpeg2FourCCControl (p. 844)": enum,
  "csigAtom (p. 844)": enum,
  "clapAtom (p. 844)": enum
},
"f4vSettings (p. 795)": {
  "moovPlacement (p. 807)": enum
},
"captionDescriptions (p. 856)": [
  { "languageDescription (p. 791)": "string",
  "destinationSettings (p. 791)": {
    "burninDestinationSettings (p. 792)": {
      "xPosition (p. 787)": integer,
      "backgroundColor (p. 788)": enum,
      "teletextSpacing (p. 788)": enum,
      "yPosition (p. 788)": integer,
      "backgroundOpacity (p. 788)": integer,
      "fontOpacity (p. 788)": integer,
      "fontResolution (p. 788)": integer,
      "shadowOpacity (p. 789)": integer,
      "shadowYOffset (p. 789)": integer,
      "outlineSize (p. 789)": integer,
      "outlineColor (p. 789)": enum,
      "fontSize (p. 789)": integer,
      "shadowXOffset (p. 789)": integer,
      "alignment (p. 790)": enum,
      "shadowColor (p. 790)": enum,
      "fontColor (p. 790)": enum
    },
    "teletextDestinationSettings (p. 792)": {
      "pageNumber (p. 861)": "string"
    },
    "ttmlDestinationSettings (p. 792)": {
      "pageNumber (p. 861)": "string"
    }
  }
}
"stylePassthrough (p. 862)": enum,
"destinationType (p. 792)": enum,
"dvbSubDestinationSettings (p. 792)": {
  "xPosition (p. 797)": integer,
  "backgroundColor (p. 798)": enum,
  "teletextSpacing (p. 798)": enum,
  "yPosition (p. 798)": integer,
  "backgroundOpacity (p. 798)": integer,
  "fontOpacity (p. 798)": integer,
  "fontResolution (p. 798)": integer,
  "shadowOpacity (p. 799)": integer,
  "shadowYOffset (p. 799)": integer,
  "outlineSize (p. 799)": integer,
  "outlineColor (p. 799)": enum,
  "framerate (p. 861)": enum
},
"sccDestinationSettings (p. 792)": {
  "framerate (p. 861)": enum
},
"languageCode (p. 791)": enum,
"customLanguageCode (p. 792)": "string"
}

Example DELETE

{
  "name (p. 796)": "string"
}

Response Bodies

Example GetPresetResponse

{
  "preset (p. 808)": {
    "createdAt (p. 855)": "string",
    "lastUpdated (p. 855)": "string",
    "settings (p. 855)": {
      "videoDescription (p. 856)": {
        "fixedAfd (p. 864)": integer,
        "scalingBehavior (p. 864)": enum,
        "respondToAfd (p. 864)": enum,
        "gopSizeUnits (p. 820)": enum,
        "flickerAdaptiveQuantization (p. 820)": enum,
        "gopSizeUnits (p. 820)": enum,
        "hrdBufferSize (p. 820)": integer,
        "hrdBufferSize (p. 820)": integer,
        "shadowXOffset (p. 799)": integer,
        "shadowColor (p. 800)": enum,
        "foregroundColor (p. 800)": enum,
      }
    }
  }
}
"qualityTuningLevel (p. 820)": enum,
"maxBitrate (p. 820)": integer,
"bitrate (p. 821)": integer,
"spatialAdaptiveQuantization (p. 821)": enum,
"maxBitrate (p. 821)": integer,
"sampleAdaptiveOffsetFilterMode (p. 821)": enum,
"numberOfIds (p. 821)": enum,
"slowPal (p. 821)": enum,
"tiles (p. 821)": enum,
"writeMpeg4PackagingType (p. 821)": enum,
"codecProfile (p. 821)": enum,
"alternateTransferFunctionSei (p. 821)": enum,
"unregisteredSeiTimecode (p. 822)": enum,
"framerateControl (p. 822)": enum,
"telecine (p. 822)": enum,
"codecLevel (p. 822)": enum,
"framerateConversionAlgorithm (p. 822)": enum,
"framerateConversionAlgorithm (p. 822)": integer,
"numberIdReferenceFrames (p. 822)": integer,
"temporalAdaptiveQuantization (p. 822)": enum,
"hrdBufferInitialFillPercentage (p. 822)": integer,
"framerateDenominator (p. 822)": integer,
"gopClosedCadence (p. 823)": integer,
"numberBFramesBetweenReferenceFrames (p. 823)": integer,
"framerateDenominator (p. 823)": integer,
"adaptiveQuantization (p. 823)": enum,
"interlaceMode (p. 823)": enum,
"gopSize (p. 823)": number,
"gopBReference (p. 824)": integer,
"framerateControl (p. 824)": enum,
"telecine (p. 824)": enum,
"framerateConversionAlgorithm (p. 824)": enum,
"adaptiveQuantization (p. 824)": enum,
"framerateConversionAlgorithm (p. 824)": integer,
"framerateControl (p. 824)": integer,
"parDenominator (p. 824)": integer,
"framerateControl (p. 824)": integer,
"framerateControl (p. 824)": enum,
"rateControlMode (p. 824)": enum
},
"codec (p. 864)": enum,
"proresSettings (p. 864)": {
    "framerateDenominator (p. 857)": integer,
    "framerateControl (p. 858)": enum,
    "telecine (p. 858)": enum,
    "framerateConversionAlgorithm (p. 858)": enum,
    "interlaceMode (p. 858)": enum,
    "gopSize (p. 858)": number,
    "gopBReference (p. 859)": integer,
    "framerateControl (p. 859)": integer,
    "parDenominator (p. 859)": integer,
    "framerateControl (p. 859)": integer
},
"mpeg2Settings (p. 864)": {
    "minInterval (p. 848)": integer,
    "parNumerator (p. 848)": integer,
    "gopSizeUnits (p. 848)": integer,
    "hrdBufferSize (p. 849)": integer,
    "framerateConversionAlgorithm (p. 849)": enum,
    "maxBitrate (p. 849)": integer,
    "bitrate (p. 849)": integer,
    "framerateConversionAlgorithm (p. 849)": enum,
    "framerateConversionAlgorithm (p. 849)": integer,
"gopClosedCadence (p. 851)": integer,
"numberBFramesBetweenReferenceFrames (p. 851)": integer,
"framerateDenominator (p. 851)": integer,
"adaptiveQuantization (p. 851)": enum,
"framerateConversionAlgorithm (p. 851)": enum,
"gopSize (p. 851)": number,
"entropyEncoding (p. 851)": enum,
"fieldEncoding (p. 851)": enum,
"adaptiveQuantization (p. 851)": enum,
"interlaceMode (p. 851)": enum,
"gopClosedCadence (p. 852)": integer,
"numberBFramesBetweenReferenceFrames (p. 852)": integer,
"framerateDenominator (p. 852)": integer,
"adaptiveQuantization (p. 852)": enum,
"framerateConversionAlgorithm (p. 852)": enum,
"gopSize (p. 852)": number,
"entropyEncoding (p. 852)": enum,
"fieldEncoding (p. 852)": enum,
"adaptiveQuantization (p. 852)": enum,
"rateControlMode (p. 852)": enum
},
"frameCaptureSettings (p. 864)": {
  "framerateDenominator (p. 807)": integer,
  "maxCaptures (p. 807)": integer,
  "framerateNumerator (p. 807)": integer,
  "quality (p. 807)": integer
},
"h264Settings (p. 864)": {
  "slices (p. 811)": integer,
  "minIInterval (p. 811)": integer,
  "parNumerator (p. 811)": integer,
  "flickerAdaptiveQuantization (p. 811)": enum,
  "gopSizeUnits (p. 811)": enum,
  "hrdBufferSize (p. 811)": integer,
  "qualityTuningLevel (p. 812)": enum,
  "maxBitrate (p. 812)": integer,
  "bitrate (p. 812)": integer,
  "spatialAdaptiveQuantization (p. 812)": enum,
  "slowPal (p. 812)": integer,
  "codecProfile (p. 812)": enum,
  "unregisteredSeiTimecode (p. 812)": enum,
  "softness (p. 812)": integer,
  "framerateControl (p. 813)": enum,
  "telecine (p. 813)": enum,
  "codecLevel (p. 813)": enum,
  "framerateConversionAlgorithm (p. 813)": enum,
  "numberReferenceFrames (p. 813)": integer,
  "temporalAdaptiveQuantization (p. 813)": enum,
  "repeatPps (p. 813)": integer,
  "hrdBufferInitialFillPercentage (p. 813)": integer,
  "framerateNumerator (p. 814)": integer,
  "gopClosedCadence (p. 814)": integer,
  "numberBFramesBetweenReferenceFrames (p. 814)": integer,
  "framerateDenominator (p. 814)": integer,
  "entropyEncoding (p. 814)": enum,
  "fieldEncoding (p. 814)": enum,
  "adaptiveQuantization (p. 814)": enum,
  "interlaceMode (p. 815)": enum,
  "gopSize (p. 815)": number,
  "gopBReference (p. 815)": enum,
  "syntax (p. 815)": enum,
  "parNumerator (p. 815)": integer,
  "sceneChangeDetect (p. 815)": enum,
  "parControl (p. 815)": enum,
  "rateControlMode (p. 815)": enum
}
},
"afdSignaling (p. 865)": enum,
"colorMetadata (p. 865)": enum,
"timecodeInsertion (p. 865)": enum,
"width (p. 865)": integer,
"videoPreprocessors (p. 865)": {
  "noiseReducer (p. 866)": {
    "filter (p. 853)": enum,
    "filterSettings (p. 853)":

"strength (p. 853)": integer,
"spatialFilterSettings (p. 853)": {
  "strength (p. 854)": integer,
  "postFilterSharpenStrength (p. 854)": integer,
  "speed (p. 854)": integer
},
"timecodeBurnin (p. 866)": {
  "prefix (p. 861)": "string",
  "fontSize (p. 861)": integer,
  "position (p. 862)": enum
},
"colorCorrector (p. 866)": {
  "saturation (p. 793)": integer,
  "brightness (p. 793)": integer,
  "hdr10Metadata (p. 793)": {
    "redPrimaryY (p. 825)": integer,
    "greenPrimaryY (p. 826)": integer,
    "whitePointX (p. 826)": integer,
    "maxLuminance (p. 826)": integer,
    "greenPrimaryX (p. 826)": integer,
    "whitePointY (p. 826)": integer,
    "redPrimaryX (p. 827)": integer,
    "bluePrimaryX (p. 827)": integer,
    "bluePrimaryY (p. 827)": integer,
    "maxFrameAverageLightLevel (p. 827)": integer,
    "maxContentLightLevel (p. 827)": integer,
    "minLuminance (p. 828)": integer
  },
  "contrast (p. 793)": integer,
  "hue (p. 793)": integer,
  "colorSpaceConversion (p. 794)": enum
},
"deinterlacer (p. 867)": {
  "mode (p. 795)": enum,
  "control (p. 795)": enum,
  "algorithm (p. 795)": enum
},
"imageInserter (p. 867)": {
  "insertableImages (p. 828)": [
    {
      "duration (p. 828)": integer,
      "fadeOut (p. 828)": integer,
      "imageY (p. 828)": integer,
      "fadeIn (p. 828)": integer,
      "imageX (p. 829)": integer,
      "width (p. 829)": integer,
      "startTime (p. 829)": "string",
      "opacity (p. 829)": integer,
      "layer (p. 829)": integer,
      "height (p. 830)": integer,
      "imageInserterInput (p. 830)": "string"
    }
  ]
},
"antiAlias (p. 865)": enum,
"position (p. 865)": {
  "width (p. 859)": integer,
  "x (p. 859)": integer,
  "y (p. 859)": integer,
  "height (p. 860)": integer
},
"sharpness (p. 865)": integer,
"crop (p. 866)": {
"width (p. 859)" : integer,
"x (p. 859)" : integer,
"y (p. 859)" : integer,
"height (p. 860)" : integer,
"dropFrameTimecode (p. 866)" : enum,
"height (p. 866)" : integer,
"audioDescriptions (p. 856)" : [
  {
    "audioTypeControl (p. 784)" : enum,
    "languageCodeControl (p. 784)" : enum,
    "remixSettings (p. 784)" : {
      "channelsOut (p. 860)" : integer,
      "channelMapping (p. 860)" : {
        "outputChannels (p. 793)" : [
          "inputChannels (p. 854)" : [integer]
        ]
      },
      "channelsIn (p. 860)" : integer,
    },
    "audioType (p. 784)" : integer,
    "audioSourceName (p. 785)" : "string",
    "codecSettings (p. 785)" : {
      "codec (p. 783)" : enum,
      "wavSettings (p. 783)" : {
        "channels (p. 867)" : integer,
        "bitDepth (p. 867)" : integer,
        "format (p. 868)" : enum,
        "sampleRate (p. 868)" : integer
      },
      "aacSettings (p. 784)" : {
        "vbrQuality (p. 779)" : enum,
        "codecProfile (p. 779)" : enum,
        "codingMode (p. 779)" : enum,
        "specification (p. 779)" : enum,
        "bitrate (p. 779)" : integer,
        "rawFormat (p. 779)" : enum,
        "rateControlMode (p. 780)" : enum,
        "sampleRate (p. 780)" : integer,
        "audioDescriptionBroadcasterMix (p. 780)" : enum
      },
      "ac3Settings (p. 784)" : {
        "dynamicRangeCompressionProfile (p. 781)" : enum,
        "dialnorm (p. 781)" : integer,
        "codingMode (p. 781)" : enum,
        "metadataControl (p. 781)" : enum,
        "bitrate (p. 782)" : integer,
        "lfeFilter (p. 782)" : enum,
        "bitstreamMode (p. 782)" : enum,
        "sampleRate (p. 782)" : integer
      },
      "siffSettings (p. 784)" : {
        "channels (p. 782)" : integer,
        "bitDepth (p. 783)" : integer,
        "sampleRate (p. 783)" : integer
      },
      "eac3Settings (p. 784)" : {
        "dialnorm (p. 803)" : integer,
        "passthroughControl (p. 803)" : enum,
        "metadataControl (p. 803)" : enum,
        "bitrate (p. 803)" : integer,
      }
    },
"dynamicRangeCompressionRf (p. 804)": enum,
"sampleRate (p. 804)": integer,
"surroundExMode (p. 804)": enum,
"ltRtSurroundMixLevel (p. 804)": number,
"dynamicRangeCompressionLine (p. 804)": enum,
"lfeControl (p. 804)": enum,
"codingMode (p. 804)": enum,
"surroundMode (p. 804)": enum,
"attenuationControl (p. 805)": enum,
"lfeFilter (p. 805)": enum,
"phaseControl (p. 805)": enum,
"ltRtCenterMixLevel (p. 805)": number,
"dcFilter (p. 805)": enum,
"bitstreamMode (p. 805)": enum,
"stereoDownmix (p. 805)": enum,
"loRoSurroundMixLevel (p. 805)": number,
"loRoCenterMixLevel (p. 806)" : number
},
"mp2Settings (p. 784)" : {
  "channels (p. 844)" : integer,
  "bitrate (p. 845)" : integer,
  "sampleRate (p. 845)" : integer
},
"languageCode (p. 785)" : enum,
"streamName (p. 785)" : "string",
"audioNormalizationSettings (p. 785)" : {
  "targetLkfs (p. 786)" : number,
  "algorithmControl (p. 787)" : enum,
  "loudnessLogging (p. 787)" : enum,
  "peakCalculation (p. 787)" : enum,
  "correctionGateLevel (p. 787)" : integer,
  "algorithm (p. 787)" : enum
},
"customLanguageCode (p. 785)" : "string",
},
"containerSettings (p. 856)" : {
  "container (p. 794)" : enum,
  "mp4Settings (p. 794)" : {
    "mp4MajorBrand (p. 845)" : "string",
    "moovPlacement (p. 846)" : enum,
    "cs1gAtom (p. 846)" : enum,
    "freeSpaceBox (p. 846)" : enum
  },
  "m3u8Settings (p. 794)" : {
    "pmtPid (p. 841)" : integer,
    "nielsen1d3 (p. 841)" : enum,
    "pcrPId (p. 841)" : integer,
    "audioPids (p. 841)" : [integer,
    "audioFramesPerPes (p. 841)" : integer,
    "scte35PId (p. 841)" : integer,
    "transportStreamId (p. 842)" : integer,
    "pcrControl (p. 842)" : enum,
    "videoPId (p. 842)" : integer,
    "privateMetadataPId (p. 842)" : integer,
    "pmtInterval (p. 842)" : integer,
    "patInterval (p. 842)" : integer,
    "programNumber (p. 843)" : integer,
    "timedMetadataPId (p. 843)" : integer,
    "timedMetadata (p. 843)" : enum,
    "scte35Source (p. 843)" : enum
  },
  "m2tsSettings (p. 794)" : {
"dvbTeletextPid (p. 835)": integer,
"nielsenId3 (p. 836)": enum,
"bitrate (p. 836)": integer,
"segmentationTime (p. 836)": number,
"audioPids (p. 836)": [
  integer
],
"rateMode (p. 836)": enum,
"maxPcrInterval (p. 836)": integer,
"audioFramesPerPes (p. 836)": integer,
"ebpAudioInterval (p. 837)": enum,
"fragmentTime (p. 837)": number,
"scte35Pid (p. 837)": integer,
"pmtInterval (p. 837)": integer,
"privateMetadataPid (p. 837)": integer,
"segmentationStyle (p. 837)": enum,
"audioBufferModel (p. 837)": enum,
"programNumber (p. 838)": integer,
"dvbNitSettings (p. 838)": {
  "networkName (p. 796)": "string",
  "networkId (p. 796)": integer,
  "nitInterval (p. 797)": integer
},
"scte35Source (p. 838)": enum,
"pmtPid (p. 838)": integer,
"bufferModel (p. 838)": enum,
"ebpPlacement (p. 838)": enum,
"dvbSdtSettings (p. 838)": {
  "sdtInterval (p. 797)": integer,
  "serviceName (p. 797)": "string",
  "serviceProviderName (p. 797)": "string",
  "outputSdt (p. 797)": enum
},
"nullPacketBitrate (p. 839)": number,
"pcrPid (p. 839)": integer,
"minEbpInterval (p. 839)": integer,
"transportStreamId (p. 839)": integer,
"videoPid (p. 839)": integer,
"pcrControl (p. 840)": enum,
"esRateInPes (p. 840)": enum,
"segmentationMarkers (p. 840)": enum,
"dvbTdtSettings (p. 840)": {
  "tdtInterval (p. 801)": integer
},
"patInterval (p. 840)": integer,
"dvbSubPids (p. 840)": [
  integer
],
"movSettings (p. 794)": {
  "reference (p. 844)": enum,
  "paddingControl (p. 844)": enum,
  "mpeg2FourCCControl (p. 844)": enum,
  "cslgAtom (p. 844)": enum,
  "clapAtom (p. 844)": enum
},
"f4vSettings (p. 795)": {
  "moovPlacement (p. 807)": enum
},
"captionDescriptions (p. 856)": [
  {
    "languageDescription (p. 791)": "string",
    "destinationSettings (p. 791)": {
      "burninDestinationSettings (p. 792)": {

```
"xPosition (p. 787)" : integer,
"backgroundColor (p. 788)" : enum,
"teletextSpacing (p. 788)" : enum,
"yPosition (p. 788)" : integer,
"backgroundOpacity (p. 788)" : integer,
"fontOpacity (p. 788)" : integer,
"fontResolution (p. 788)" : integer,
"shadowOpacity (p. 789)" : integer,
"shadowYOffset (p. 789)" : integer,
"outlineSize (p. 789)" : integer,
"outlineColor (p. 789)" : enum,
"fontSize (p. 789)" : integer,
"shadowXOffset (p. 789)" : integer,
"alignment (p. 790)" : enum,
"shadowColor (p. 790)" : enum,
"fontColor (p. 790)" : enum
},
"teletextDestinationSettings (p. 792)" : {
  "pageNumber (p. 861)" : "string"
},
"ttmlDestinationSettings (p. 792)" : {
  "stylePassthrough (p. 862)" : enum
},
"destinationType (p. 792)" : enum,
"dvbSubDestinationSettings (p. 792)" : {
  "xPosition (p. 797)" : integer,
  "backgroundColor (p. 798)" : enum,
  "teletextSpacing (p. 798)" : enum,
  "yPosition (p. 798)" : integer,
  "backgroundOpacity (p. 798)" : integer,
  "fontOpacity (p. 798)" : integer,
  "fontResolution (p. 798)" : integer,
  "shadowOpacity (p. 799)" : integer,
  "shadowYOffset (p. 799)" : integer,
  "outlineSize (p. 799)" : integer,
  "outlineColor (p. 799)" : enum,
  "fontSize (p. 799)" : integer,
  "shadowXOffset (p. 799)" : integer,
  "alignment (p. 800)" : enum,
  "shadowColor (p. 800)" : enum,
  "fontColor (p. 800)" : enum
},
"sccDestinationSettings (p. 792)" : {
  "framerate (p. 861)" : enum
},
"languageCode (p. 791)" : enum,
"customLanguageCode (p. 792)" : "string"
}

Example UpdatePresetResponse

{
  "preset (p. 863)" : {
    "createdAt (p. 855)" : "string",
  


```
770
```
"lastUpdated (p. 855)": "string",
"settings (p. 855)": {
  "videoDescription (p. 856)": {
    "fixedAfd (p. 864)": integer,
    "scalingBehavior (p. 864)": enum,
    "respondToAfd (p. 864)": enum,
    "codecSettings (p. 864)": {
      "h265Settings (p. 863)": {
        "slices (p. 819)": integer,
        "minIInterval (p. 820)": integer,
        "parNumerator (p. 820)": integer,
        "flickerAdaptiveQuantization (p. 820)": enum,
        "gopSizeUnits (p. 820)": enum,
        "hrdBufferSize (p. 820)": integer,
        "qualityTuningLevel (p. 820)": enum,
        "maxBitrate (p. 820)": integer,
        "bitrate (p. 821)": integer,
        "spatialAdaptiveQuantization (p. 821)": enum,
        "sampleAdaptiveOffsetFilterMode (p. 821)": enum,
        "temporalIds (p. 821)": enum,
        "slowPal (p. 821)": enum,
        "tiles (p. 821)": enum,
        "writeMp4PackagingType (p. 821)": enum,
        "codecProfile (p. 821)": enum,
        "alternateTransferFunctionSel (p. 821)": enum,
        "unregisteredSelTimecode (p. 822)": enum,
        "framerateControl (p. 822)": enum,
        "telecine (p. 822)": enum,
        "codecLevel (p. 822)": enum,
        "framerateConversionAlgorithm (p. 822)": enum,
        "numberReferenceFrames (p. 822)": integer,
        "temporalAdaptiveQuantization (p. 822)": enum,
        "hrdBufferInitialFillPercentage (p. 822)": integer,
        "framerateNumerator (p. 822)": integer,
        "framerateDenominator (p. 823)": integer
      }
    }
  }
}
"proresSettings (p. 864)": {
  "framerateDenominator (p. 857)": integer,
  "slowPal (p. 858)": enum,
  "framerateControl (p. 858)": enum,
  "telecine (p. 858)": enum,
  "framerateConversionAlgorithm (p. 858)": enum,
  "interlaceMode (p. 858)": enum,
  "parNumerator (p. 858)": integer,
  "codecProfile (p. 858)": enum,
  "parDenominator (p. 858)": integer,
  "framerateNumerator (p. 859)": integer
}
"mpeg2Settings (p. 864)": {
  "minIInterval (p. 848)": integer,
  "parNumerator (p. 848)": integer,
  "gopSizeUnits (p. 848)": enum,
  "hrdBufferSize (p. 849)": integer,
  "qualityTuningLevel (p. 849)": enum,
"maxBitrate (p. 849)": integer,
"bitrate (p. 849)": integer,
"spatialAdaptiveQuantization (p. 849)": enum,
"slowPal (p. 849)": enum,
"intraDcPrecision (p. 849)": enum,
"codecProfile (p. 849)": enum,
"softness (p. 850)": integer,
"framerateControl (p. 850)": enum,
"telecine (p. 850)": enum,
"codecLevel (p. 850)": enum,
"framerateConversionAlgorithm (p. 850)": enum,
"temporalAdaptiveQuantization (p. 850)": enum,
"hrdBufferInitialFillPercentage (p. 850)": integer,
"framerateNumerator (p. 850)": integer,
"gopClosedCadence (p. 851)": integer,
"numberBFramesBetweenReferenceFrames (p. 851)": integer,
"framerateDenominator (p. 851)": integer,
"adaptiveQuantization (p. 851)": enum,
"gopSize (p. 851)": number,
"syntax (p. 851)": enum,
"parDenominator (p. 852)": integer,
"sceneChangeDetect (p. 852)": enum,
"parControl (p. 852)": enum,
"rateControlMode (p. 852)": enum
},
"frameCaptureSettings (p. 864)": {
  "framerateDenominator (p. 807)": integer,
  "maxCaptures (p. 807)": integer,
  "framerateNumerator (p. 807)": integer,
  "quality (p. 807)": integer
},
"h264Settings (p. 864)": {
  "slices (p. 811)": integer,
  "minIInterval (p. 811)": integer,
  "parNumerator (p. 811)": integer,
  "flickerAdaptiveQuantization (p. 811)": enum,
  "gopSizeUnits (p. 811)": enum,
  "hrdBufferSize (p. 811)": integer,
  "qualityTuningLevel (p. 812)": enum,
  "maxBitrate (p. 812)": integer,
  "bitrate (p. 812)": integer,
  "spatialAdaptiveQuantization (p. 812)": enum,
  "slowPal (p. 812)": enum,
  "codecProfile (p. 812)": enum,
  "unregisteredSeiTimecode (p. 812)": enum,
  "softness (p. 812)": integer,
  "framerateControl (p. 813)": enum,
  "telecine (p. 813)": enum,
  "codecLevel (p. 813)": enum,
  "framerateConversionAlgorithm (p. 813)": enum,
  "numberReferenceFrames (p. 813)": integer,
  "temporalAdaptiveQuantization (p. 813)": enum,
  "repeatPps (p. 813)": enum,
  "hrdBufferInitialFillPercentage (p. 813)": integer,
  "framerateNumerator (p. 814)": integer,
  "gopClosedCadence (p. 814)": integer,
  "numberBFramesBetweenReferenceFrames (p. 814)": integer,
  "framerateDenominator (p. 814)": integer,
  "entropyEncoding (p. 814)": enum,
  "fieldEncoding (p. 814)": enum,
  "adaptiveQuantization (p. 815)": enum,
  "interlaceMode (p. 815)": enum,
  "gopSize (p. 815)": number,
  "gop8Reference (p. 815)": enum,
  "syntax (p. 815)": enum,
"parDenominator (p. 815)": integer,
"sceneChangeDetect (p. 815)": enum,
"parControl (p. 815)": enum,
"rateControlMode (p. 815)": enum
},
"afdSignaling (p. 865)": enum,
"colorMetadata (p. 865)": enum,
"timecodeInsertion (p. 865)": enum,
"width (p. 865)": integer,
"videoPreprocessors (p. 865)": {
  "noiseReducer (p. 866)": {
    "filter (p. 853)": enum,
    "filterSettings (p. 853)": {
      "strength (p. 853)": integer
    }
  },
  "spatialFilterSettings (p. 853)": {
    "strength (p. 854)": integer,
    "postFilterSharpenStrength (p. 854)": integer,
    "speed (p. 854)": integer
  }
},
"timecodeBurnin (p. 866)": {
  "prefix (p. 861)": "string",
  "fontSize (p. 861)": integer,
  "position (p. 862)": enum
},
"colorCorrector (p. 866)": {
  "saturation (p. 793)": integer,
  "brightness (p. 793)": integer,
  "hdr10Metadata (p. 793)": {
    "redPrimaryY (p. 825)": integer,
    "greenPrimaryY (p. 826)": integer,
    "whitePointX (p. 826)": integer,
    "maxLuminance (p. 826)": integer,
    "greenPrimaryX (p. 826)": integer,
    "whitePointY (p. 826)": integer,
    "redPrimaryX (p. 827)": integer,
    "bluePrimaryX (p. 827)": integer,
    "bluePrimaryY (p. 827)": integer,
    "maxFrameAverageLightLevel (p. 827)": integer,
    "maxContentLightLevel (p. 827)": integer,
    "minLuminance (p. 828)": integer
  },
  "contrast (p. 793)": integer,
  "hue (p. 793)": integer,
  "colorSpaceConversion (p. 794)": enum
},
"deinterlacer (p. 867)": {
  "mode (p. 795)": enum,
  "control (p. 795)": enum,
  "algorithm (p. 795)": enum
},
"imageInserter (p. 867)": {
  "insertableImages (p. 828)": [
    {
      "duration (p. 828)": integer,
      "fadeOut (p. 828)": integer,
      "imageY (p. 828)": integer,
      "fadeIn (p. 828)": integer,
      "imageX (p. 828)": integer,
      "width (p. 829)": integer,
      "startTime (p. 829)": "string",
      "opacity (p. 829)": integer,
      "layer (p. 829)": integer,
      "height (p. 830)": integer,
  }]}
"imageInserterInput (p. 830)": "string"

"antiAlias (p. 865)": enum,
"position (p. 865)": {
  "width (p. 859)": integer,
  "x (p. 859)": integer,
  "y (p. 859)": integer,
  "height (p. 860)": integer
},
"sharpness (p. 865)": integer,
"crop (p. 866)": {
  "width (p. 859)": integer,
  "x (p. 859)": integer,
  "y (p. 859)": integer,
  "height (p. 860)": integer
},
"dropFrameTimecode (p. 866)": enum,
"height (p. 866)": integer
},
"audioDescriptions (p. 856)": [
{
  "audioTypeControl (p. 784)": enum,
  "languageCodeControl (p. 784)": enum,
  "remixSettings (p. 784)": {
    "channelOut (p. 860)": integer,
    "channelMapping (p. 860)": {
      "outputChannels (p. 793)": [ 
        "inputChannels (p. 854)": [ integer
      ]
    }
  }
},
"channelsIn (p. 860)": integer
],
"audioType (p. 784)": integer,
"audioSourceName (p. 785)": "string",
"codecSettings (p. 785)": {
  "codec (p. 783)": enum,
  "wavSettings (p. 783)": {
    "channels (p. 867)": integer,
    "bitDepth (p. 867)": integer,
    "format (p. 868)": enum,
    "sampleRate (p. 868)": integer
  },
  "aacSettings (p. 784)": {
    "vbrQuality (p. 779)": enum,
    "codecProfile (p. 779)": enum,
    "codingMode (p. 779)": enum,
    "specification (p. 779)": enum,
    "bitrate (p. 779)": integer,
    "rawFormat (p. 779)": enum,
    "rateControlMode (p. 780)": enum,
    "sampleRate (p. 780)": integer,
    "audioDescriptionBroadcasterMix (p. 780)": enum
  },
  "ac3Settings (p. 784)": {
    "dynamicRangeCompressionProfile (p. 781)": enum,
    "dialnorm (p. 781)": integer,
    "codingMode (p. 781)": enum,
    "metadataControl (p. 781)": enum,
    "bitrateControl (p. 781)": enum,
    "sampleRate (p. 782)": integer,
    "metadataControl (p. 781)": enum,
    "bitrateControl (p. 781)": enum,
    "sampleRate (p. 782)": integer,
"lfeFilter (p. 782)": enum,
"bitstreamMode (p. 782)": enum,
"sampleRate (p. 782)": integer
},
"aiffSettings (p. 784)": {
  "channels (p. 782)": integer,
  "bitDepth (p. 783)": integer,
  "sampleRate (p. 783)": integer
},
"eac3Settings (p. 784)": {
  "dialnorm (p. 803)": integer,
  "passthroughControl (p. 803)": enum,
  "metadataControl (p. 803)": enum,
  "bitrate (p. 803)": integer,
  "dynamicRangeCompressionRf (p. 804)": enum,
  "sampleRate (p. 804)": integer,
  "surroundExMode (p. 804)": enum,
  "lfeControl (p. 804)": enum,
  "codingMode (p. 804)": enum,
  "surroundMode (p. 804)": enum,
  "attenuationControl (p. 805)": enum,
  "lfeFilter (p. 805)": enum,
  "phaseControl (p. 805)": enum,
  "lRtCenterMixLevel (p. 805)": number,
  "dcFilter (p. 805)": enum,
  "bitstreamMode (p. 805)": enum,
  "stereoDownmix (p. 805)": enum,
  "loRoSurroundMixLevel (p. 805)": number,
  "loRoCenterMixLevel (p. 806)": number
},
"mp2Settings (p. 784)": {
  "channels (p. 844)": integer,
  "bitrate (p. 845)": integer,
  "sampleRate (p. 845)": integer
},
"languageCode (p. 785)": enum,
"streamName (p. 785)": "string",
"audioNormalizationSettings (p. 785)": {
  "targetLkfs (p. 786)": number,
  "algorithmControl (p. 787)": enum,
  "loudnessLogging (p. 787)": enum,
  "peakCalculation (p. 787)": enum,
  "correctionGateLevel (p. 787)": integer,
  "algorithm (p. 787)": enum
},
"customLanguageCode (p. 785)": "string",
"containerSettings (p. 856)": {
  "container (p. 794)": enum,
  "mp4Settings (p. 794)": {
    "mp4MajorBrand (p. 845)": "string",
    "moovPlacement (p. 846)": enum,
    "csigAtom (p. 846)": enum,
    "freeSpaceBox (p. 846)": enum
  },
  "m3u8Settings (p. 794)": {
    "pmtPid (p. 841)": integer,
    "nielsenId3 (p. 841)": enum,
    "pcrPid (p. 841)": integer,
    "audioPids (p. 841)": [ integer
    ],
<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;audioFramesPerPes (p. 841)&quot;</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>&quot;scte35Pid (p. 841)&quot;</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>&quot;transportStreamId (p. 842)&quot;</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>&quot;pcrControl (p. 842)&quot;</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>&quot;videoPid (p. 842)&quot;</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>&quot;privateMetadataPid (p. 842)&quot;</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>&quot;pmtInterval (p. 842)&quot;</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>&quot;patInterval (p. 842)&quot;</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>&quot;programNumber (p. 843)&quot;</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>&quot;timedMetadataPid (p. 843)&quot;</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>&quot;timedMetadata (p. 843)&quot;</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>&quot;scte35Source (p. 843)&quot;</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>&quot;m2tsSettings (p. 794)&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;dvbTeletextPid (p. 835)&quot;</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>&quot;nielsenId3 (p. 836)&quot;</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>&quot;bitrate (p. 836)&quot;</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>&quot;segmentationTime (p. 836)&quot;</td>
<td>number</td>
<td></td>
</tr>
<tr>
<td>&quot;audioPids (p. 836)&quot;</td>
<td>[integer]</td>
<td></td>
</tr>
<tr>
<td>&quot;rateMode (p. 836)&quot;</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>&quot;maxPcrInterval (p. 836)&quot;</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>&quot;audioFramesPerPes (p. 836)&quot;</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>&quot;ebpAudioInterval (p. 837)&quot;</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>&quot;fragmentTime (p. 837)&quot;</td>
<td>number</td>
<td></td>
</tr>
<tr>
<td>&quot;scte35Pid (p. 837)&quot;</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>&quot;pmtInterval (p. 837)&quot;</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>&quot;privateMetadataPid (p. 837)&quot;</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>&quot;segmentationStyle (p. 837)&quot;</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>&quot;audioBufferModel (p. 837)&quot;</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>&quot;programNumber (p. 838)&quot;</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>&quot;dvbNitSettings (p. 838)&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;networkName (p. 796)&quot;</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>&quot;networkId (p. 796)&quot;</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>&quot;nitInterval (p. 797)&quot;</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>&quot;timedMetadataPid (p. 838)&quot;</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>&quot;scte35Source (p. 838)&quot;</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>&quot;bufferModel (p. 838)&quot;</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>&quot;ebpPlacement (p. 838)&quot;</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>&quot;dvbSdtSettings (p. 838)&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;sdtInterval (p. 797)&quot;</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>&quot;serviceName (p. 797)&quot;</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>&quot;serviceProviderName (p. 797)&quot;</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>&quot;outputSdt (p. 797)&quot;</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>&quot;nullPacketBitrate (p. 839)&quot;</td>
<td>number</td>
<td></td>
</tr>
<tr>
<td>&quot;pcrPid (p. 839)&quot;</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>&quot;minEbpInterval (p. 839)&quot;</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>&quot;transportStreamId (p. 839)&quot;</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>&quot;videoPid (p. 839)&quot;</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>&quot;pcrControl (p. 840)&quot;</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>&quot;esRateInPes (p. 840)&quot;</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>&quot;segmentationMarkers (p. 840)&quot;</td>
<td>enum</td>
<td></td>
</tr>
<tr>
<td>&quot;dvbTdtSettings (p. 840)&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;tdtInterval (p. 801)&quot;</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>&quot;patInterval (p. 840)&quot;</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>&quot;dvbSubPids (p. 840)&quot;</td>
<td>[integer]</td>
<td></td>
</tr>
<tr>
<td>&quot;movSettings (p. 794)&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot;reference (p. 844)&quot;</td>
<td>enum</td>
<td></td>
</tr>
</tbody>
</table>
"paddingControl (p. 844)\": enum,
"mpeg2FourCCControl (p. 844)\": enum,
"cslgAtom (p. 844)\": enum,
"clapAtom (p. 844)\": enum
},
"f4vSettings (p. 795)\": {
"moovPlacement (p. 807)\": enum
},
"captionDescriptions (p. 856)\": [
{
"languageDescription (p. 791)\": "string",
"destinationSettings (p. 791)\": {
"burninDestinationSettings (p. 792)\": {
"xPosition (p. 787)\": integer,
"backgroundColor (p. 788)\": enum,
"teletextSpacing (p. 788)\": enum,
"yPosition (p. 788)\": integer,
"backgroundOpacity (p. 788)\": integer,
"fontOpacity (p. 788)\": integer,
"fontResolution (p. 788)\": integer,
"shadowOpacity (p. 789)\": integer,
"shadowYOffset (p. 789)\": integer,
"outlineSize (p. 789)\": integer,
"outlineColor (p. 789)\": enum,
"fontSize (p. 789)\": integer,
"shadowXOffset (p. 789)\": integer,
"alignment (p. 790)\": enum,
"shadowColor (p. 790)\": enum,
"fontColor (p. 790)\": enum
},
"teletextDestinationSettings (p. 792)\": {
&pageNumber (p. 861)\": "string"
},
"ttmlDestinationSettings (p. 792)\": {
"stylePassthrough (p. 862)\": enum
},
"destinationType (p. 792)\": enum,
"dvbSubDestinationSettings (p. 792)\": {
"xPosition (p. 797)\": integer,
"backgroundColor (p. 798)\": enum,
"teletextSpacing (p. 798)\": enum,
"yPosition (p. 798)\": integer,
"backgroundOpacity (p. 798)\": integer,
"fontOpacity (p. 798)\": integer,
"fontResolution (p. 798)\": integer,
"shadowOpacity (p. 799)\": integer,
"shadowYOffset (p. 799)\": integer,
"outlineSize (p. 799)\": integer,
"outlineColor (p. 799)\": enum,
"fontSize (p. 799)\": integer,
"shadowXOffset (p. 799)\": integer,
"alignment (p. 800)\": enum,
"shadowColor (p. 800)\": enum,
"fontColor (p. 800)\": enum
},
"sccDestinationSettings (p. 792)\": {
"framerate (p. 861)\": enum
}
}

"languageCode (p. 791)\": enum,
"customLanguageCode (p. 792)\": "string"
}

"name (p. 855)\": "string",
Properties

**AacAudioDescriptionBroadcasterMix (enum)**

Choose BROADCASTER_MIXED_AD when the input contains pre-mixed main audio + audio description (AD) as a stereo pair. The value for AudioType will be set to 3, which signals to downstream systems that this stream contains "broadcaster mixed AD". Note that the input received by the encoder must contain pre-mixed audio; the encoder does not perform the mixing. When you choose BROADCASTER_MIXED_AD, the encoder ignores any values you provide in AudioType and FollowInputAudioType. Choose NORMAL when the input does not contain pre-mixed audio + audio description (AD). In this case, the encoder will use any values you provide for AudioType and FollowInputAudioType.

- BROADCASTER_MIXED_AD
- NORMAL

**AacCodecProfile (enum)**

AAC Profile.

- LC
- HEV1
- HEV2

**AacCodingMode (enum)**

Mono (Audio Description), Mono, Stereo, or 5.1 channel layout. Valid values depend on rate control mode and profile. "1.0 - Audio Description (Receiver Mix)" setting receives a stereo description plus control track and emits a mono AAC encode of the description track, with control data emitted in the PES header as per ETSI TS 101 154 Annex E.

- AD_RECEIVER_MIX
- CODING_MODE_1_0
- CODING_MODE_1_1
- CODING_MODE_2_0
CODING_MODE_5_1

AacRateControlMode (enum)
Rate Control Mode.
  CBR
  VBR

AacRawFormat (enum)
Enables LATM/LOAS AAC output. Note that if you use LATM/LOAS AAC in an output, you must choose "No container" for the output container.
  LATM_LOAS
  NONE

AacSettings

vbrQuality
  Type: string
  Required: False

codecProfile
  Type: string
  Required: False

codingMode
  Type: string
  Required: True

specification
  Type: string
  Required: False

bitrate
Average bitrate in bits/second. Defaults and valid values depend on rate control mode and profile.
  Type: integer
  Required: False
  Minimum: 6000
  Maximum: 1024000

rawFormat
  Type: string
  Required: False
rateControlMode

Type: string
Required: False

sampleRate

Sample rate in Hz. Valid values depend on rate control mode and profile.

Type: integer
Required: True
Minimum: 8000
Maximum: 96000

audioDescriptionBroadcasterMix

Type: string
Required: False

AacSpecification (enum)

Use MPEG-2 AAC instead of MPEG-4 AAC audio for raw or MPEG-2 Transport Stream containers.

MPEG2
MPEG4

AacVbrQuality (enum)

VBR Quality Level - Only used if rate_control_mode is VBR.

LOW
MEDIUM_LOW
MEDIUM_HIGH
HIGH

Ac3BitstreamMode (enum)

Specifies the "Bitstream Mode" (bsmod) for the emitted AC-3 stream. See ATSC A/52-2012 for background on these values.

COMPLETE_MAIN
COMMENTARY
DIALOGUE
EMERGENCY
HEARING_IMPAIRED
MUSIC_AND_EFFECTS
VISUALLY_IMPAIRED
VOICE_OVER

Ac3CodingMode (enum)

Dolby Digital coding mode. Determines number of channels.

CODING_MODE_1_0
CODING_MODE_1_1
CODING_MODE_2_0
CODING_MODE_3_2_LFE

**Ac3DynamicRangeCompressionProfile (enum)**

If set to FILM_STANDARD, adds dynamic range compression signaling to the output bitstream as defined in the Dolby Digital specification.

- FILM_STANDARD
- NONE

**Ac3LfeFilter (enum)**

 Applies a 120Hz lowpass filter to the LFE channel prior to encoding. Only valid with 3_2_LFE coding mode.

- ENABLED
- DISABLED

**Ac3MetadataControl (enum)**

When set to FOLLOW_INPUT, encoder metadata will be sourced from the DD, DD+, or DolbyE decoder that supplied this audio data. If audio was not supplied from one of these streams, then the static metadata settings will be used.

- FOLLOW_INPUT
- USE_CONFIGURED

**Ac3Settings**

**dynamicRangeCompressionProfile**

- **Type:** string
- **Required:** False

**dialnorm**

Sets the dialnorm for the output. If blank and input audio is Dolby Digital, dialnorm will be passed through.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 31

**codingMode**

- **Type:** string
- **Required:** False

**metadataControl**

- **Type:** string
Required: False

bitrate
Average bitrate in bits/second. Valid bitrates depend on the coding mode.

Type: integer
Required: False
Minimum: 64000
Maximum: 640000

lfeFilter

Type: string
Required: False

bitstreamMode

Type: string
Required: False

sampleRate
Sample rate in hz. Sample rate is always 48000.

Type: integer
Required: False
Minimum: 48000
Maximum: 48000

AfdSignaling (enum)
This setting only applies to H.264 and MPEG2 outputs. Use Insert AFD signaling (AfdSignaling) to specify whether the service includes AFD values in the output video data and what those values are. * Choose None to remove all AFD values from this output. * Choose Fixed to ignore input AFD values and instead encode the value specified in the job. * Choose Auto to calculate output AFD values based on the input AFD scaler data.

NONE
AUTO
FIXED

AiffSettings

channels
Set Channels to specify the number of channels in this output audio track. Choosing Mono in the console will give you 1 output channel; choosing Stereo will give you 2. In the API, valid values are 1 and 2.

Type: integer
Required: False
Minimum: 1
Maximum: 2

**bitDepth**

Specify Bit depth (BitDepth), in bits per sample, to choose the encoding quality for this audio track.

*Type:* integer  
*Required:* False  
*Minimum:* 16  
*Maximum:* 24

**sampleRate**

Sample rate in hz.

*Type:* integer  
*Required:* False  
*Minimum:* 8000  
*Maximum:* 192000

**AntiAlias (enum)**

Enable Anti-alias (AntiAlias) to enhance sharp edges in video output when your input resolution is much larger than your output resolution. Default is enabled.

- DISABLED
- ENABLED

**AudioCodec (enum)**

Type of Audio codec.

- AAC
- MP2
- WAV
- AIFF
- AC3
- EAC3
- PASSTHROUGH

**AudioCodecSettings**

**codec**

*Type:* string  
*Required:* True

**wavSettings**

*Type:* WavSettings (p. 867)  
*Required:* False
aacSettings
  Type: AacSettings (p. 779)
  Required: False

ac3Settings
  Type: Ac3Settings (p. 781)
  Required: False

aiffSettings
  Type: AiffSettings (p. 782)
  Required: False

eac3Settings
  Type: Eac3Settings (p. 803)
  Required: False

mp2Settings
  Type: Mp2Settings (p. 844)
  Required: False

AudioDescription

audioTypeControl
  Type: string
  Required: False

languageCodeControl
  Type: string
  Required: False

remixSettings
  Advanced audio remixing settings.
  Type: RemixSettings (p. 860)
  Required: False

audioType
  Applies only if Follow Input Audio Type is unchecked (false). A number between 0 and 255. The following
  are defined in ISO-IEC 13818-1: 0 = Undefined, 1 = Clean Effects, 2 = Hearing Impaired, 3 = Visually
  Impaired Commentary, 4-255 = Reserved.
  Type: integer
  Required: False
Properties

Minimum: 0
Maximum: 255

**audioSourceName**

Specifies which audio data to use from each input. In the simplest case, specify an "Audio Selector":#inputs-audio_selector by name based on its order within each input. For example if you specify "Audio Selector 3", then the third audio selector will be used from each input. If an input does not have an "Audio Selector 3", then the audio selector marked as "default" in that input will be used. If there is no audio selector marked as "default", silence will be inserted for the duration of that input. Alternatively, an "Audio Selector Group":#inputs-audio_selector_group name may be specified, with similar default/silence behavior. If no audio_source_name is specified, then "Audio Selector 1" will be chosen automatically.

Type: string
Required: False

**codecSettings**

Type: AudioCodecSettings (p. 783)
Required: True

**languageCode**

Indicates the language of the audio output track. The ISO 639 language specified in the 'Language Code' drop down will be used when 'Follow Input Language Code' is not selected or when 'Follow Input Language Code' is selected but there is no ISO 639 language code specified by the input.

Type: string
Required: False

**streamName**

Used for MS Smooth and Apple HLS outputs. Indicates the name displayed by the player (eg. English, or Director Commentary). Alphanumeric characters, spaces, and underscore are legal.

Type: string
Required: False
Pattern: ^\w\s*$

**audioNormalizationSettings**

Type: AudioNormalizationSettings (p. 786)
Required: False

**customLanguageCode**

Specify the language for this audio output track, using the ISO 639-2 or ISO 639-3 three-letter language code. The language specified will be used when 'Follow Input Language Code' is not selected or when 'Follow Input Language Code' is selected but there is no ISO 639 language code specified by the input.

Type: string
Required: False
**Pattern:** `^[A-Za-z]{3}$`

**AudioLanguageCodeControl (enum)**

Choosing `FOLLOW_INPUT` will cause the ISO 639 language code of the output to follow the ISO 639 language code of the input. The language specified for `languageCode` will be used when `USE_CONFIGURED` is selected or when `FOLLOW_INPUT` is selected but there is no ISO 639 language code specified by the input.

- `FOLLOW_INPUT`
- `USE_CONFIGURED`

**AudioNormalizationAlgorithm (enum)**

Audio normalization algorithm to use. 1770-1 conforms to the CALM Act specification, 1770-2 conforms to the EBU R-128 specification.

- `ITU_BS_1770_1`
- `ITU_BS_1770_2`

**AudioNormalizationAlgorithmControl (enum)**

When enabled the output audio is corrected using the chosen algorithm. If disabled, the audio will be measured but not adjusted.

- `CORRECT_AUDIO`
- `MEASURE_ONLY`

**AudioNormalizationLoudnessLogging (enum)**

If set to `LOG`, log each output's audio track loudness to a CSV file.

- `LOG`
- `DONT_LOG`

**AudioNormalizationPeakCalculation (enum)**

If set to `TRUE_PEAK`, calculate and log the TruePeak for each output's audio track loudness.

- `TRUE_PEAK`
- `NONE`

**AudioNormalizationSettings**

**targetLkfs**

Target LKFS(loudness) to adjust volume to. If no value is entered, a default value will be used according to the chosen algorithm. The CALM Act (1770-1) recommends a target of -24 LKFS. The EBU R-128 specification (1770-2) recommends a target of -23 LKFS.

- **Type**: number
- **Required**: False
- **Format**: float
- **Minimum**: -59.0
Maximum: 0.0

algorithmControl

Type: string
Required: False

loudnessLogging

Type: string
Required: False

peakCalculation

Type: string
Required: False

correctionGateLevel

Content measuring above this level will be corrected to the target level. Content measuring below this level will not be corrected. Gating only applies when not using real_time_correction.

Type: integer
Required: False
Minimum: -70
Maximum: 0

algorithm

Type: string
Required: False

AudioTypeControl (enum)

When set to FOLLOW_INPUT, if the input contains an ISO 639 audio_type, then that value is passed through to the output. If the input contains no ISO 639 audio_type, the value in Audio Type is included in the output. Otherwise the value in Audio Type is included in the output. Note that this field and audioType are both ignored if audioDescriptionBroadcasterMix is set to BROADCASTER_MIXED_AD.

FOLLOW_INPUT
USE_CONFIGURED

BurninDestinationSettings

xPosition

Specifies the horizontal position of the caption relative to the left side of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the left of the output. If no explicit x_position is provided, the horizontal caption position will be determined by the alignment parameter. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

Type: integer
**BackgroundColor**

- **Type:** string
- **Required:** False

**TeletextSpacing**

- **Type:** string
- **Required:** False

**YPosition**

Specifies the vertical position of the caption relative to the top of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the top of the output. If no explicit y_position is provided, the caption will be positioned towards the bottom of the output. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
- **Maximum:** 2147483647

**BackgroundOpacity**

Specifies the opacity of the background rectangle. 255 is opaque; 0 is transparent. Leaving this parameter blank is equivalent to setting it to 0 (transparent). All burn-in and DVB-Sub font settings must match.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
- **Maximum:** 255

**FontOpacity**

Specifies the opacity of the burned-in captions. 255 is opaque; 0 is transparent. All burn-in and DVB-Sub font settings must match.

- **Type:** integer
- **Required:** True
- **Minimum:** 0
- **Maximum:** 255

**FontResolution**

Font resolution in DPI (dots per inch); default is 96 dpi. All burn-in and DVB-Sub font settings must match.

- **Type:** integer
shadowOpacity

Specifies the opacity of the shadow. 255 is opaque; 0 is transparent. Leaving this parameter blank is equivalent to setting it to 0 (transparent). All burn-in and DVB-Sub font settings must match.

Type: integer
Required: False
Minimum: 0
Maximum: 255

shadowYOffset

Specifies the vertical offset of the shadow relative to the captions in pixels. A value of -2 would result in a shadow offset 2 pixels above the text. All burn-in and DVB-Sub font settings must match.

Type: integer
Required: False
Minimum: -2147483648
Maximum: 2147483647

outlineSize

Specifies font outline size in pixels. This option is not valid for source captions that are either 608/ embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

Type: integer
Required: True
Minimum: 0
Maximum: 10

outlineColor

Type: string
Required: True

fontSize

A positive integer indicates the exact font size in points. Set to 0 for automatic font size selection. All burn-in and DVB-Sub font settings must match.

Type: integer
Required: False
Minimum: 0
Maximum: 96

shadowXOffset

Specifies the horizontal offset of the shadow relative to the captions in pixels. A value of -2 would result in a shadow offset 2 pixels to the left. All burn-in and DVB-Sub font settings must match.
Type: integer  
Required: False  
Minimum: -2147483648  
Maximum: 2147483647

**alignment**

Type: string  
Required: True

**shadowColor**

Type: string  
Required: False

**fontColor**

Type: string  
Required: False

**BurninSubtitleAlignment (enum)**

If no explicit x_position or y_position is provided, setting alignment to centered will place the captions at the bottom center of the output. Similarly, setting a left alignment will align captions to the bottom left of the output. If x and y positions are given in conjunction with the alignment parameter, the font will be justified (either left or centered) relative to those coordinates. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

CENTERED  
LEFT

**BurninSubtitleBackgroundColor (enum)**

Specifies the color of the rectangle behind the captions. All burn-in and DVB-Sub font settings must match.

NONE  
BLACK  
WHITE

**BurninSubtitleFontColor (enum)**

Specifies the color of the burned-in captions. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

WHITE  
BLACK  
YELLOW  
RED  
GREEN  
BLUE
**BurninSubtitleOutlineColor (enum)**

Specifies font outline color. This option is not valid for source captions that are either 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- BLACK
- WHITE
- YELLOW
- RED
- GREEN
- BLUE

**BurninSubtitleShadowColor (enum)**

Specifies the color of the shadow cast by the captions. All burn-in and DVB-Sub font settings must match.

- NONE
- BLACK
- WHITE

**BurninSubtitleTeletextSpacing (enum)**

Only applies to jobs with input captions in Teletext or STL formats. Specify whether the spacing between letters in your captions is set by the captions grid or varies depending on letter width. Choose fixed grid to conform to the spacing specified in the captions file more accurately. Choose proportional to make the text easier to read if the captions are closed caption.

- FIXED_GRID
- PROPORTIONAL

**CaptionDescriptionPreset**

**languageDescription**

Human readable information to indicate captions available for players (eg. English, or Spanish). Alphanumeric characters, spaces, and underscore are legal.

- **Type**: string
- **Required**: False
- **Pattern**: `^[\w ]*$`

**destinationSettings**

- **Type**: CaptionDestinationSettings (p. 792)
- **Required**: True

**languageCode**

Indicates the language of the caption output track.

- **Type**: string
- **Required**: False
customLanguageCode

Indicates the language of the caption output track, using the ISO 639-2 or ISO 639-3 three-letter language code

Type: string
Required: False
Pattern: ^[A-Za-z]{3}$

CaptionDestinationSettings

burninDestinationSettings

Type: BurninDestinationSettings (p. 787)
Required: False

teletextDestinationSettings

Type: TeletextDestinationSettings (p. 861)
Required: False

ttmlDestinationSettings

Type: TtmlDestinationSettings (p. 862)
Required: False

destinationType

Type: string
Required: True

dvbSubDestinationSettings

Type: DvbSubDestinationSettings (p. 797)
Required: False

sccDestinationSettings

Type: SccDestinationSettings (p. 861)
Required: False

CaptionDestinationType (enum)

Type of Caption output, including Burn-In, Embedded, SCC, SRT, TTML, WebVTT, DVB-Sub, Teletext.

BURN_IN
DVB_SUB
EMBEDDED
SCC
SRT
TELETEXT
TTML
WEBVTT

ChannelMapping

outputChannels

List of output channels

Type: Array of type OutputChannelMapping (p. 854)
Required: True

ColorCorrector

saturation

Saturation level.

Type: integer
Required: False
Minimum: 1
Maximum: 100

brightness

Brightness level.

Type: integer
Required: False
Minimum: 1
Maximum: 100

hdr10Metadata

Type: Hdr10Metadata (p. 825)
Required: False

contrast

Contrast level.

Type: integer
Required: False
Minimum: 1
Maximum: 100

hue

Hue in degrees.

Type: integer
Required: False
Minimum: -180
Maximum: 180
colorSpaceConversion

Type: string  
Required: False

ColorMetadata (enum)

Enable Insert color metadata (ColorMetadata) to include color metadata in this output. This setting is enabled by default.

IGNORE  
INSERT

ColorSpaceConversion (enum)

Determines if colorspace conversion will be performed. If set to _None_, no conversion will be performed. If _Force 601_ or _Force 709_ are selected, conversion will be performed for inputs with differing colorspaces. An input's colorspace can be specified explicitly in the "Video Selector":#inputs-video-selector if necessary.

NONE  
FORCE_601  
FORCE_709  
FORCE_HDR10  
FORCE_HLG_2020

ContainerSettings

container

Type: string  
Required: True

mp4Settings

Type: Mp4Settings (p. 845)  
Required: False

m3u8Settings

Type: M3u8Settings (p. 841)  
Required: False

m2tsSettings

Type: M2tsSettings (p. 835)  
Required: False

movSettings

Type: MovSettings (p. 844)  
Required: False
f4vSettings

Type: F4vSettings (p. 807)
Required: False

ContainerType (enum)
Container for this output. Some containers require a container settings object. If not specified, the default object will be created.

F4V
ISMV
M2TS
M3U8
CMFC
MOV
MP4
MPD
MXF
RAW

DeinterlaceAlgorithm (enum)
Only applies when you set Deinterlace (DeinterlaceMode) to Deinterlace (DEINTERLACE) or Adaptive (ADAPTIVE). Motion adaptive interpolate (INTERPOLATE) produces sharper pictures, while blend (BLEND) produces smoother motion. Use (INTERPOLATE_TICKER) OR (BLEND_TICKER) if your source file includes a ticker, such as a scrolling headline at the bottom of the frame.

INTERPOLATE
INTERPOLATE_TICKER
BLEND
BLEND_TICKER

Deinterlacer

mode

Type: string
Required: False

control

Type: string
Required: False

algorithm

Type: string
Required: False

DeinterlacerControl (enum)
- When set to NORMAL (default), the deinterlacer does not convert frames that are tagged in metadata as progressive. It will only convert those that are tagged as some other type. - When set to
FORCE_ALL_FRAMES, the deinterlacer converts every frame to progressive - even those that are already tagged as progressive. Turn Force mode on only if there is a good chance that the metadata has tagged frames as progressive when they are not progressive. Do not turn on otherwise; processing frames that are already progressive into progressive will probably result in lower quality video.

FORCE_ALL_FRAMES
NORMAL

DeinterlacerMode (enum)

Use Deinterlacer (DeinterlaceMode) to choose how the service will do deinterlacing. Default is Deinterlace. - Deinterlace converts interlaced to progressive. - Inverse telecine converts Hard Telecine 29.97i to progressive 23.976p. - Adaptive auto-detects and converts to progressive.

DEINTERLACE
INVERSE_TELECINE
ADAPTIVE

DeletePresetRequest

name
The name of the preset to be deleted.

Type: string
Required: False

DeletePresetResponse

DropFrameTimecode (enum)

Applies only to 29.97 fps outputs. When this feature is enabled, the service will use drop-frame timecode on outputs. If it is not possible to use drop-frame timecode, the system will fall back to non-drop-frame. This setting is enabled by default when Timecode insertion (TimecodeInsertion) is enabled.

DISABLED
ENABLED

DvbNitSettings

networkName
The network name text placed in the network_name_descriptor inside the Network Information Table. Maximum length is 256 characters.

Type: string
Required: True

networkId
The numeric value placed in the Network Information Table (NIT).

Type: integer
Required: True
Properties

Minimum: 0
Maximum: 65535

**nitInterval**

The number of milliseconds between instances of this table in the output transport stream.

- **Type**: integer
- **Required**: True
- **Minimum**: 25
- **Maximum**: 10000

**DvbSdtSettings**

**sdInterval**

The number of milliseconds between instances of this table in the output transport stream.

- **Type**: integer
- **Required**: False
- **Minimum**: 25
- **Maximum**: 2000

**serviceName**

The service name placed in the service_descriptor in the Service Description Table. Maximum length is 256 characters.

- **Type**: string
- **Required**: False

**serviceProviderName**

The service provider name placed in the service_descriptor in the Service Description Table. Maximum length is 256 characters.

- **Type**: string
- **Required**: False

**outputSdt**

- **Type**: string
- **Required**: False

**DvbSubDestinationSettings**

**xPosition**

Specifies the horizontal position of the caption relative to the left side of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the left of the output. If no explicit x_position is provided, the horizontal caption position will be determined by the alignment parameter. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.
Properties

Type: integer
Required: False
Minimum: 0
Maximum: 2147483647

backgroundColor

Type: string
Required: False

teletextSpacing

Type: string
Required: False

tyPosition

Specifies the vertical position of the caption relative to the top of the output in pixels. A value of 10 would result in the captions starting 10 pixels from the top of the output. If no explicit y_position is provided, the caption will be positioned towards the bottom of the output. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

Type: integer
Required: False
Minimum: 0
Maximum: 2147483647

backgroundOpacity

Specifies the opacity of the background rectangle. 255 is opaque; 0 is transparent. Leaving this parameter blank is equivalent to setting it to 0 (transparent). All burn-in and DVB-Sub font settings must match.

Type: integer
Required: False
Minimum: 0
Maximum: 255

fontOpacity

Specifies the opacity of the burned-in captions. 255 is opaque; 0 is transparent. All burn-in and DVB-Sub font settings must match.

Type: integer
Required: True
Minimum: 0
Maximum: 255

fontResolution

Font resolution in DPI (dots per inch); default is 96 dpi. All burn-in and DVB-Sub font settings must match.
Properties

**shadowOpacity**

Specifies the opacity of the shadow. 255 is opaque; 0 is transparent. Leaving this parameter blank is equivalent to setting it to 0 (transparent). All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: False
- **Minimum**: 96
- **Maximum**: 600

**shadowYOffset**

Specifies the vertical offset of the shadow relative to the captions in pixels. A value of -2 would result in a shadow offset 2 pixels above the text. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: False
- **Minimum**: -2147483648
- **Maximum**: 2147483647

**outlineSize**

Specifies font outline size in pixels. This option is not valid for source captions that are either 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: True
- **Minimum**: 0
- **Maximum**: 10

**outlineColor**

- **Type**: string
- **Required**: True

**fontSize**

A positive integer indicates the exact font size in points. Set to 0 for automatic font size selection. All burn-in and DVB-Sub font settings must match.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 96

**shadowXOffset**

Specifies the horizontal offset of the shadow relative to the captions in pixels. A value of -2 would result in a shadow offset 2 pixels to the left. All burn-in and DVB-Sub font settings must match.
Type: integer  
Required: False  
Minimum: -2147483648  
Maximum: 2147483647

alignment

Type: string  
Required: True

shadowColor

Type: string  
Required: False

fontColor

Type: string  
Required: False

DvbSubtitleAlignment (enum)

If no explicit x_position or y_position is provided, setting alignment to centered will place the captions at the bottom center of the output. Similarly, setting a left alignment will align captions to the bottom left of the output. If x and y positions are given in conjunction with the alignment parameter, the font will be justified (either left or centered) relative to those coordinates. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

CENTERED  
LEFT

DvbSubtitleBackgroundColor (enum)

Specifies the color of the rectangle behind the captions. All burn-in and DVB-Sub font settings must match.

NONE  
BLACK  
WHITE

DvbSubtitleFontColor (enum)

Specifies the color of the burned-in captions. This option is not valid for source captions that are STL, 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

WHITE  
BLACK  
YELLOW  
RED  
GREEN  
BLUE
**DvbSubtitleOutlineColor (enum)**

Specifies font outline color. This option is not valid for source captions that are either 608/embedded or teletext. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

BLACK
WHITE
YELLOW
RED
GREEN
BLUE

**DvbSubtitleShadowColor (enum)**

Specifies the color of the shadow cast by the captions. All burn-in and DVB-Sub font settings must match.

NONE
BLACK
WHITE

**DvbSubtitleTeletextSpacing (enum)**

Only applies to jobs with input captions in Teletext or STL formats. Specify whether the spacing between letters in your captions is set by the captions grid or varies depending on letter width. Choose fixed grid to conform to the spacing specified in the captions file more accurately. Choose proportional to make the text easier to read if the captions are closed caption.

FIXED_GRID
PROPORTIONAL

**DvbTdtSettings**

**tdtInterval**

The number of milliseconds between instances of this table in the output transport stream.

- **Type**: integer
- **Required**: True
- **Minimum**: 1000
- **Maximum**: 30000

**Eac3AttenuationControl (enum)**

If set to ATTENUATE_3_DB, applies a 3 dB attenuation to the surround channels. Only used for 3/2 coding mode.

ATTENUATE_3_DB
NONE

**Eac3BitstreamMode (enum)**

Specifies the "Bitstream Mode" (bsmod) for the emitted E-AC-3 stream. See ATSC A/52-2012 (Annex E) for background on these values.
Eac3CodingMode (enum)
Dolby Digital Plus coding mode. Determines number of channels.

CODING_MODE_1_0
CODING_MODE_2_0
CODING_MODE_3_2

Eac3DcFilter (enum)
Activates a DC highpass filter for all input channels.

ENABLED
DISABLED

Eac3DynamicRangeCompressionLine (enum)
Enables Dynamic Range Compression that restricts the absolute peak level for a signal.

NONE
FILM_STANDARD
FILM_LIGHT
MUSIC_STANDARD
MUSIC_LIGHT
SPEECH

Eac3DynamicRangeCompressionRf (enum)
Enables Heavy Dynamic Range Compression, ensures that the instantaneous signal peaks do not exceed specified levels.

NONE
FILM_STANDARD
FILM_LIGHT
MUSIC_STANDARD
MUSIC_LIGHT
SPEECH

Eac3LfeControl (enum)
When encoding 3/2 audio, controls whether the LFE channel is enabled

LFE
NO_LFE

Eac3LfeFilter (enum)
Applies a 120Hz lowpass filter to the LFE channel prior to encoding. Only valid with 3_2_LFE coding mode.
ENABLED
DISABLED

Eac3MetadataControl (enum)
When set to FOLLOW_INPUT, encoder metadata will be sourced from the DD, DD+, or DolbyE decoder that supplied this audio data. If audio was not supplied from one of these streams, then the static metadata settings will be used.

FOLLOW_INPUT
USE_CONFIGURED

Eac3PassthroughControl (enum)
When set to WHEN_POSSIBLE, input DD+ audio will be passed through if it is present on the input. This detection is dynamic over the life of the transcode. Inputs that alternate between DD+ and non-DD+ content will have a consistent DD+ output as the system alternates between passthrough and encoding.

WHEN_POSSIBLE
NO_PASSTHROUGH

Eac3PhaseControl (enum)
Controls the amount of phase-shift applied to the surround channels. Only used for 3/2 coding mode.

SHIFT_90_DEGREES
NO_SHIFT

Eac3Settings

dialnorm
Sets the dialnorm for the output. If blank and input audio is Dolby Digital Plus, dialnorm will be passed through.

Type: integer
Required: False
Minimum: 1
Maximum: 31

passthroughControl
Type: string
Required: False

metadataControl
Type: string
Required: False

bitrate
Average bitrate in bits/second. Valid bitrates depend on the coding mode.
Properties

Type: integer
Required: False
Minimum: 64000
Maximum: 640000

dynamicRangeCompressionRf
Type: string
Required: False

sampleRate
Sample rate in hz. Sample rate is always 48000.
Type: integer
Required: False
Minimum: 48000
Maximum: 48000

surroundExMode
Type: string
Required: False

ltRtSurroundMixLevel
Left total/Right total surround mix level. Only used for 3/2 coding mode. Valid values: -1.5 -3.0 -4.5 -6.0 -60
Type: number
Required: False
Format: float
Minimum: -60.0
Maximum: -1.5

dynamicRangeCompressionLine
Type: string
Required: False

lfeControl
Type: string
Required: False

codingMode
Type: string
Required: False

surroundMode
Type: string
Required: False

attenuationControl

Type: string
Required: False

lfeFilter

Type: string
Required: False

phaseControl

Type: string
Required: False

ltRtCenterMixLevel

Left total/Right total center mix level. Only used for 3/2 coding mode. Valid values: 3.0, 1.5, 0.0, -1.5
-3.0 -4.5 -6.0 -60

Type: number
Required: False
Format: float
Minimum: -60.0
Maximum: 3.0

dcFilter

Type: string
Required: False

bitstreamMode

Type: string
Required: False

stereoDownmix

Type: string
Required: False

loRoSurroundMixLevel

Left only/Right only surround mix level. Only used for 3/2 coding mode. Valid values: -1.5 -3.0 -4.5 -6.0
-60

Type: number
Required: False
Format: float
Minimum: -60.0
Maximum: -1.5

**IoRoCenterMixLevel**

Left only/Right only center mix level. Only used for 3/2 coding mode. Valid values: 3.0, 1.5, 0.0, -1.5 -3.0 -4.5 -6.0 -60

- **Type**: number
- **Required**: False
- **Format**: float
- **Minimum**: -60.0
- **Maximum**: 3.0

**Eac3StereoDownmix (enum)**

Stereo downmix preference. Only used for 3/2 coding mode.

- NOT_INDICATED
- LO_RO
- LT_RT
- DPL2

**Eac3SurroundExMode (enum)**

When encoding 3/2 audio, sets whether an extra center back surround channel is matrix encoded into the left and right surround channels.

- NOT_INDICATED
- ENABLED
- DISABLED

**Eac3SurroundMode (enum)**

When encoding 2/0 audio, sets whether Dolby Surround is matrix encoded into the two channels.

- NOT_INDICATED
- ENABLED
- DISABLED

**ExceptionBody**

**message**

- **Type**: string
- **Required**: False

**F4vMoovPlacement (enum)**

If set to PROGRESSIVE_DOWNLOAD, the MOOV atom is relocated to the beginning of the archive as required for progressive downloading. Otherwise it is placed normally at the end.

- PROGRESSIVE_DOWNLOAD
F4vSettings

moovPlacement

Type: string
Required: False

FrameCaptureSettings

framerateDenominator

Frame capture will encode the first frame of the output stream, then one frame every framerateDenominator/framerateNumerator seconds. For example, settings of framerateNumerator = 1 and framerateDenominator = 3 (a rate of 1/3 frame per second) will capture the first frame, then 1 frame every 3s. Files will be named as filename.n.jpg where n is the 0-based sequence number of each Capture.

Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

maxCaptures

Maximum number of captures (encoded jpg output files).

Type: integer
Required: False
Minimum: 1
Maximum: 10000000

framerateNumerator

Frame capture will encode the first frame of the output stream, then one frame every framerateDenominator/framerateNumerator seconds. For example, settings of framerateNumerator = 1 and framerateDenominator = 3 (a rate of 1/3 frame per second) will capture the first frame, then 1 frame every 3s. Files will be named as filename.NNNNNNN.jpg where N is the 0-based frame sequence number zero padded to 7 decimal places.

Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

quality

JPEG Quality - a higher value equals higher quality.

Type: integer
Required: False
Minimum: 1
Maximum: 100
GetPresetRequest

name

The name of the preset.

Type: string
Required: False

GetPresetResponse

preset

Type: Preset (p. 855)
Required: False

H264AdaptiveQuantization (enum)

Adaptive quantization. Allows intra-frame quantizers to vary to improve visual quality.

OFF
LOW
MEDIUM
HIGH
HIGHER
MAX

H264CodecLevel (enum)

H.264 Level.

AUTO
LEVEL_1
LEVEL_1_1
LEVEL_1_2
LEVEL_1_3
LEVEL_2
LEVEL_2_1
LEVEL_2_2
LEVEL_3
LEVEL_3_1
LEVEL_3_2
LEVEL_4
LEVEL_4_1
LEVEL_4_2
LEVEL_5
LEVEL_5_1
LEVEL_5_2

H264CodecProfile (enum)

H.264 Profile. High 4:2:2 and 10-bit profiles are only available with the AVC-I License.

BASELINE
**Properties**

**HIGH**
**HIGH_10BIT**
**HIGH_422**
**HIGH_422_10BIT**
**MAIN**

**H264EntropyEncoding (enum)**

Entropy encoding mode. Use CABAC (must be in Main or High profile) or CAVLC.

- **CABAC**
- **CAVLC**

**H264FieldEncoding (enum)**

Choosing FORCE_FIELD disables PAFF encoding for interlaced outputs.

- **PAFF**
- **FORCE_FIELD**

**H264FlickerAdaptiveQuantization (enum)**

Adjust quantization within each frame to reduce flicker or 'pop' on I-frames.

- **DISABLED**
- **ENABLED**

**H264FramerateControl (enum)**

If you are using the console, use the Framerate setting to specify the framerate for this output. If you want to keep the same framerate as the input video, choose Follow source. If you want to do framerate conversion, choose a framerate from the dropdown list or choose Custom. The framerates shown in the dropdown list are decimal approximations of fractions. If you choose Custom, specify your framerate as a fraction. If you are creating your transcoding job specification as a JSON file without the console, use FramerateControl to specify which value the service uses for the framerate for this output. Choose INITIALIZE_FROM_SOURCE if you want the service to use the framerate from the input. Choose SPECIFIED if you want the service to use the framerate you specify in the settings FramerateNumerator and FramerateDenominator.

- **INITIALIZE_FROM_SOURCE**
- **SPECIFIED**

**H264FramerateConversionAlgorithm (enum)**

When set to INTERPOLATE, produces smoother motion during framerate conversion.

- **DUPLICATE_DROP**
- **INTERPOLATE**

**H264GopBReference (enum)**

If enable, use reference B frames for GOP structures that have B frames > 1.

- **DISABLED**
- **ENABLED**
H264GopSizeUnits (enum)

Indicates if the GOP Size in H264 is specified in frames or seconds. If seconds the system will convert the GOP Size into a frame count at run time.

FRAMES
SECONDS

H264InterlaceMode (enum)

Use Interlace mode (InterlaceMode) to choose the scan line type for the output. * Top Field First (TOP_FIELD) and Bottom Field First (BOTTOM_FIELD) produce interlaced output with the entire output having the same field polarity (top or bottom first). * Follow, Default Top (FOLLOW_TOP_FIELD) and Follow, Default Bottom (FOLLOW_BOTTOM_FIELD) use the same field polarity as the source. Therefore, behavior depends on the input scan type, as follows. - If the source is interlaced, the output will be interlaced with the same polarity as the source (it will follow the source). The output could therefore be a mix of "top field first" and "bottom field first". - If the source is progressive, the output will be interlaced with "top field first" or "bottom field first" polarity, depending on which of the Follow options you chose.

PROGRESSIVE
TOP_FIELD
BOTTOM_FIELD
FOLLOW_TOP_FIELD
FOLLOW_BOTTOM_FIELD

H264ParControl (enum)

Using the API, enable ParFollowSource if you want the service to use the pixel aspect ratio from the input. Using the console, do this by choosing Follow source for Pixel aspect ratio.

INITIALIZE_FROM_SOURCE
SPECIFIED

H264QualityTuningLevel (enum)

Use Quality tuning level (H264QualityTuningLevel) to specify whether to use fast single-pass, high-quality singlepass, or high-quality multipass video encoding.

SINGLE_PASS
SINGLE_PASS_HQ
MULTI_PASS_HQ

H264RateControlMode (enum)

Use this setting to specify whether this output has a variable bitrate (VBR) or constant bitrate (CBR).

VBR
CBR

H264RepeatPps (enum)

Places a PPS header on each encoded picture, even if repeated.

DISABLED
ENABLED
H264SceneChangeDetect (enum)
Scene change detection (inserts I-frames on scene changes).

- DISABLED
- ENABLED

H264Settings

slices
Number of slices per picture. Must be less than or equal to the number of macroblock rows for progressive pictures, and less than or equal to half the number of macroblock rows for interlaced pictures.

- Type: integer
- Required: False
- Minimum: 1
- Maximum: 32

minIInterval
Enforces separation between repeated (cadence) I-frames and I-frames inserted by Scene Change Detection. If a scene change I-frame is within I-interval frames of a cadence I-frame, the GOP is shrunk and/or stretched to the scene change I-frame. GOP stretch requires enabling lookahead as well as setting I-interval. The normal cadence resumes for the next GOP. This setting is only used when Scene Change Detect is enabled. Note: Maximum GOP stretch = GOP size + Min-I-interval - 1

- Type: integer
- Required: False
- Minimum: 0
- Maximum: 30

parNumerator
Pixel Aspect Ratio numerator.

- Type: integer
- Required: False
- Minimum: 1
- Maximum: 2147483647

flickerAdaptiveQuantization

- Type: string
- Required: False

gopSizeUnits

- Type: string
- Required: False

hrdBufferSize
Size of buffer (HRD buffer model) in bits. For example, enter five megabits as 5000000.
Properties

**Type**: integer
**Required**: False
**Minimum**: 0
**Maximum**: 1152000000

**qualityTuningLevel**

**Type**: string
**Required**: False

**maxBitrate**

Maximum bitrate in bits/second. For example, enter five megabits per second as 5000000.

**Type**: integer
**Required**: False
**Minimum**: 1000
**Maximum**: 1152000000

**bitrate**

Average bitrate in bits/second. Required for VBR and CBR. For MS Smooth outputs, bitrates must be unique when rounded down to the nearest multiple of 1000.

**Type**: integer
**Required**: False
**Minimum**: 1000
**Maximum**: 1152000000

**spatialAdaptiveQuantization**

**Type**: string
**Required**: False

**slowPal**

**Type**: string
**Required**: False

**codecProfile**

**Type**: string
**Required**: False

**unregisteredSeiTimecode**

**Type**: string
**Required**: False

**softness**

Softness. Selects quantizer matrix, larger values reduce high-frequency content in the encoded image.
Properties

**framerateControl**

Type: string  
Required: False

**telecine**

Type: string  
Required: False

**codecLevel**

Type: string  
Required: False

**framerateConversionAlgorithm**

Type: string  
Required: False

**numberReferenceFrames**

Number of reference frames to use. The encoder may use more than requested if using B-frames and/or interlaced encoding.

Type: integer  
Required: False  
Minimum: 1  
Maximum: 6

**temporalAdaptiveQuantization**

Type: string  
Required: False

**repeatPps**

Type: string  
Required: False

**hrdBufferInitialFillPercentage**

Percentage of the buffer that should initially be filled (HRD buffer model).

Type: integer  
Required: False  
Minimum: 0
**Properties**

**Maximum**: 100

**framerateNumerator**

Framerate numerator - framerate is a fraction, e.g. 24000 / 1001 = 23.976 fps.

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 2147483647

**gopClosedCadence**

Frequency of closed GOPs. In streaming applications, it is recommended that this be set to 1 so a decoder joining mid-stream will receive an IDR frame as quickly as possible. Setting this value to 0 will break output segmenting.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 2147483647

**numberBFramesBetweenReferenceFrames**

Number of B-frames between reference frames.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 7

**framerateDenominator**

When you use the API for transcode jobs that use framerate conversion, specify the framerate as a fraction. For example, 24000 / 1001 = 23.976 fps. Use FramerateDenominator to specify the denominator of this fraction. In this example, use 1001 for the value of FramerateDenominator. When you use the console for transcode jobs that use framerate conversion, provide the value as a decimal number for Framerate. In this example, specify 23.976.

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 2147483647

**entropyEncoding**

- **Type**: string
- **Required**: False

**fieldEncoding**

- **Type**: string
- **Required**: False
adaptiveQuantization

Type: string
Required: False

interlaceMode

Type: string
Required: False

gopSize

GOP Length (keyframe interval) in frames or seconds. Must be greater than zero.

Type: number
Required: False
Format: float
Minimum: 0.0

gopBReference

Type: string
Required: False

syntax

Type: string
Required: False

parDenominator

Pixel Aspect Ratio denominator.

Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

sceneChangeDetect

Type: string
Required: False

parControl

Type: string
Required: False

rateControlMode

Type: string
Required: False
H264SlowPal (enum)

Enables Slow PAL rate conversion. 23.976fps and 24fps input is relabeled as 25fps, and audio is sped up correspondingly.

DISABLED
ENABLED

H264SpatialAdaptiveQuantization (enum)

Adjust quantization within each frame based on spatial variation of content complexity.

DISABLED
ENABLED

H264Syntax (enum)

Produces a bitstream compliant with SMPTE RP-2027.

DEFAULT
RP2027

H264Telecine (enum)

This field applies only if the Streams > Advanced > Framerate (framerate) field is set to 29.970. This field works with the Streams > Advanced > Preprocessors > Deinterlacer field (deinterlace_mode) and the Streams > Advanced > Interlaced Mode field (interlace_mode) to identify the scan type for the output: Progressive, Interlaced, Hard Telecine or Soft Telecine. - Hard: produces 29.97i output from 23.976 input. - Soft: produces 23.976; the player converts this output to 29.97i.

NONE
SOFT
HARD

H264TemporalAdaptiveQuantization (enum)

Adjust quantization within each frame based on temporal variation of content complexity.

DISABLED
ENABLED

H264UnregisteredSeiTimecode (enum)

Inserts timecode for each frame as 4 bytes of an unregistered SEI message.

DISABLED
ENABLED

H265AdaptiveQuantization (enum)

Adaptive quantization. Allows intra-frame quantizers to vary to improve visual quality.

OFF
LOW
**MEDIUM**
**HIGH**
**HIGHER**
**MAX**

**H265AlternateTransferFunctionSei (enum)**

Enables Alternate Transfer Function SEI message for outputs using Hybrid Log Gamma (HLG) Electro-Optical Transfer Function (EOTF).

- DISABLED
- ENABLED

**H265CodecLevel (enum)**

H.265 Level.

- AUTO
- LEVEL_1
- LEVEL_2
- LEVEL_2_1
- LEVEL_3
- LEVEL_3_1
- LEVEL_4
- LEVEL_4_1
- LEVEL_5
- LEVEL_5_1
- LEVEL_5_2
- LEVEL_6
- LEVEL_6_1
- LEVEL_6_2

**H265CodecProfile (enum)**

Represents the Profile and Tier, per the HEVC (H.265) specification. Selections are grouped as [Profile] / [Tier], so “Main/High” represents Main Profile with High Tier. 4:2:2 profiles are only available with the HEVC 4:2:2 License.

- MAIN_MAIN
- MAIN_HIGH
- MAIN10_MAIN
- MAIN10_HIGH
- MAIN_422_8BIT_MAIN
- MAIN_422_8BIT_HIGH
- MAIN_422_10BIT_MAIN
- MAIN_422_10BIT_HIGH

**H265FlickerAdaptiveQuantization (enum)**

Adjust quantization within each frame to reduce flicker or ‘pop’ on I-frames.

- DISABLED
ENABLED

**H265FramerateControl (enum)**

If you are using the console, use the Framerate setting to specify the framerate for this output. If you want to keep the same framerate as the input video, choose Follow source. If you want to do framerate conversion, choose a framerate from the dropdown list or choose Custom. The framerates shown in the dropdown list are decimal approximations of fractions. If you choose Custom, specify your framerate as a fraction. If you are creating your transcoding job specification as a JSON file without the console, use FramerateControl to specify which value the service uses for the framerate for this output. Choose INITIALIZE_FROM_SOURCE if you want the service to use the framerate from the input. Choose SPECIFIED if you want the service to use the framerate you specify in the settings FramerateNumerator and FramerateDenominator.

- INITIALIZE_FROM_SOURCE
- SPECIFIED

**H265FramerateConversionAlgorithm (enum)**

When set to INTERPOLATE, produces smoother motion during framerate conversion.

- DUPLICATE_DROP
- INTERPOLATE

**H265GopBReference (enum)**

If enable, use reference B frames for GOP structures that have B frames > 1.

- DISABLED
- ENABLED

**H265GopSizeUnits (enum)**

Indicates if the GOP Size in H265 is specified in frames or seconds. If seconds the system will convert the GOP Size into a frame count at run time.

- FRAMES
- SECONDS

**H265InterlaceMode (enum)**

Use Interlace mode (InterlaceMode) to choose the scan line type for the output. * Top Field First (TOP_FIELD) and Bottom Field First (BOTTOM_FIELD) produce interlaced output with the entire output having the same field polarity (top or bottom first). * Follow, Default Top (FOLLOW_TOP_FIELD) and Follow, Default Bottom (FOLLOW_BOTTOM_FIELD) use the same field polarity as the source. Therefore, behavior depends on the input scan type. - If the source is interlaced, the output will be interlaced with the same polarity as the source (it will follow the source). The output could therefore be a mix of “top field first” and “bottom field first”. - If the source is progressive, the output will be interlaced with “top field first” or “bottom field first” polarity, depending on which of the Follow options you chose.

- PROGRESSIVE
- TOP_FIELD
- BOTTOM_FIELD
- FOLLOW_TOP_FIELD
FOLLOW_BOTTOM_FIELD

**H265ParControl (enum)**

Using the API, enable ParFollowSource if you want the service to use the pixel aspect ratio from the input. Using the console, do this by choosing Follow source for Pixel aspect ratio.

- INITIALIZE_FROM_SOURCE
- SPECIFIED

**H265QualityTuningLevel (enum)**

Use Quality tuning level (H265QualityTuningLevel) to specify whether to use fast single-pass, high-quality singlepass, or high-quality multipass video encoding.

- SINGLE_PASS
- SINGLE_PASS_HQ
- MULTI_PASS_HQ

**H265RateControlMode (enum)**

Use this setting to specify whether this output has a variable bitrate (VBR) or constant bitrate (CBR).

- VBR
- CBR

**H265SampleAdaptiveOffsetFilterMode (enum)**

Specify Sample Adaptive Offset (SAO) filter strength. Adaptive mode dynamically selects best strength based on content.

- DEFAULT
- ADAPTIVE
- OFF

**H265SceneChangeDetect (enum)**

Scene change detection (inserts I-frames on scene changes).

- DISABLED
- ENABLED

**H265Settings**

**slices**

Number of slices per picture. Must be less than or equal to the number of macroblock rows for progressive pictures, and less than or equal to half the number of macroblock rows for interlaced pictures.

- **Type**: integer
- **Required**: False
- **Minimum**: 1
**Properties**

**minInterval**
Enforces separation between repeated (cadence) I-frames and I-frames inserted by Scene Change Detection. If a scene change I-frame is within I-interval frames of a cadence I-frame, the GOP is shrunk and/or stretched to the scene change I-frame. GOP stretch requires enabling lookahead as well as setting I-interval. The normal cadence resumes for the next GOP. This setting is only used when Scene Change Detect is enabled. Note: Maximum GOP stretch = GOP size + Min-I-interval - 1

*Type: integer*
*Required: False*
*Minimum: 0*
*Maximum: 30*

**parNumerator**
Pixel Aspect Ratio numerator.

*Type: integer*
*Required: False*
*Minimum: 1*
*Maximum: 2147483647*

**flickerAdaptiveQuantization**

*Type: string*
*Required: False*

**gopSizeUnits**

*Type: string*
*Required: False*

**hrdBufferSize**
Size of buffer (HRD buffer model) in bits. For example, enter five megabits as 5000000.

*Type: integer*
*Required: False*
*Minimum: 0*
*Maximum: 1466400000*

**qualityTuningLevel**

*Type: string*
*Required: False*

**maxBitrate**
Maximum bitrate in bits/second.

*Type: integer*
*Required: False*
**Properties**

**bitrate**

Average bitrate in bits/second. Required for VBR and CBR. For MS Smooth outputs, bitrates must be unique when rounded down to the nearest multiple of 1000.

- **Type**: integer
- **Required**: False
- **Minimum**: 1000
- **Maximum**: 1466400000

**spatialAdaptiveQuantization**

- **Type**: string
- **Required**: False

**sampleAdaptiveOffsetFilterMode**

- **Type**: string
- **Required**: False

**temporalIds**

- **Type**: string
- **Required**: False

**slowPal**

- **Type**: string
- **Required**: False

**tiles**

- **Type**: string
- **Required**: False

**writeMp4PackagingType**

- **Type**: string
- **Required**: False

**codecProfile**

- **Type**: string
- **Required**: False

**alternateTransferFunctionSei**

- **Type**: string
- **Required**: False
unregisteredseiTimecode
  Type: string
  Required: False

framerateControl
  Type: string
  Required: False

telecine
  Type: string
  Required: False

codecLevel
  Type: string
  Required: False

framerateConversionAlgorithm
  Type: string
  Required: False

numberReferenceFrames
Number of reference frames to use. The encoder may use more than requested if using B-frames and/or interlaced encoding.
  Type: integer
  Required: False
  Minimum: 1
  Maximum: 6

temporalAdaptiveQuantization
  Type: string
  Required: False

hrdBufferInitialFillPercentage
Percentage of the buffer that should initially be filled (HRD buffer model).
  Type: integer
  Required: False
  Minimum: 0
  Maximum: 100

framerateNumerator
Framerate numerator - framerate is a fraction, e.g. 24000 / 1001 = 23.976 fps.
Properties

**Type**
- Type: integer
- Required: False
- Minimum: 1
- Maximum: 2147483647

**gopClosedCadence**
Frequency of closed GOPs. In streaming applications, it is recommended that this be set to 1 so a decoder joining mid-stream will receive an IDR frame as quickly as possible. Setting this value to 0 will break output segmenting.
- Type: integer
- Required: False
- Minimum: 0
- Maximum: 2147483647

**numberBFramesBetweenReferenceFrames**
Number of B-frames between reference frames.
- Type: integer
- Required: False
- Minimum: 0
- Maximum: 7

**framerateDenominator**
Framerate denominator.
- Type: integer
- Required: False
- Minimum: 1
- Maximum: 2147483647

**adaptiveQuantization**
- Type: string
- Required: False

**interlaceMode**
- Type: string
- Required: False

**gopSize**
GOP Length (keyframe interval) in frames or seconds. Must be greater than zero.
- Type: number
- Required: False
- Format: float
- Minimum: 0.0
**gopBReference**

*Type:* string  
*Required:* False

**parDenominator**

Pixel Aspect Ratio denominator.

*Type:* integer  
*Required:* False  
*Minimum:* 1  
*Maximum:* 2147483647

**sceneChangeDetect**

*Type:* string  
*Required:* False

**parControl**

*Type:* string  
*Required:* False

**rateControlMode**

*Type:* string  
*Required:* False

**H265SlowPal (enum)**

Enables Slow PAL rate conversion. 23.976fps and 24fps input is relabeled as 25fps, and audio is sped up correspondingly.

- DISABLED  
- ENABLED

**H265SpatialAdaptiveQuantization (enum)**

Adjust quantization within each frame based on spatial variation of content complexity.

- DISABLED
- ENABLED

**H265Telecine (enum)**

This field applies only if the Streams > Advanced > Framerate (framerate) field is set to 29.970. This field works with the Streams > Advanced > Preprocessors > Deinterlacer field (deinterlace_mode) and the Streams > Advanced > Interlaced Mode field (interlace_mode) to identify the scan type for the output: Progressive, Interlaced, Hard Telecine or Soft Telecine. - Hard: produces 29.97i output from 23.976 input.  
- Soft: produces 23.976; the player converts this output to 29.97i.

- NONE  
- SOFT
HARD

**H265TemporalAdaptiveQuantization (enum)**
Adjust quantization within each frame based on temporal variation of content complexity.

- DISABLED
- ENABLED

**H265TemporalIds (enum)**
Enables temporal layer identifiers in the encoded bitstream. Up to 3 layers are supported depending on GOP structure: I- and P-frames form one layer, reference B-frames can form a second layer and non-reference b-frames can form a third layer. Decoders can optionally decode only the lower temporal layers to generate a lower frame rate output. For example, given a bitstream with temporal IDs and with b-frames = 1 (i.e. IbPbPb display order), a decoder could decode all the frames for full frame rate output or only the I and P frames (lowest temporal layer) for a half frame rate output.

- DISABLED
- ENABLED

**H265Tiles (enum)**
Enable use of tiles, allowing horizontal as well as vertical subdivision of the encoded pictures.

- DISABLED
- ENABLED

**H265UnregisteredSeiTimecode (enum)**
Inserts timecode for each frame as 4 bytes of an unregistered SEI message.

- DISABLED
- ENABLED

**H265WriteMp4PackagingType (enum)**
If HVC1, output that is H.265 will be marked as HVC1 and adhere to the ISO-IECJTC1-SC29_N13798_Text_ISOIEC_FDIS_14496-15_3rd_E spec which states that parameter set NAL units will be stored in the sample headers but not in the samples directly. If HEV1, then H.265 will be marked as HEV1 and parameter set NAL units will be written into the samples.

- HVC1
- HEV1

**Hdr10Metadata**

**redPrimaryY**
HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

- **Type**: integer
Required: False
Minimum: 0
Maximum: 50000

greenPrimaryY
HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

Type: integer
Required: False
Minimum: 0
Maximum: 50000

whitePointX
HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

Type: integer
Required: False
Minimum: 0
Maximum: 50000

maxLuminance
Nominal maximum mastering display luminance in units of of 0.0001 candelas per square meter.

Type: integer
Required: False
Minimum: 0
Maximum: 2147483647

greenPrimaryX
HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

Type: integer
Required: False
Minimum: 0
Maximum: 50000

whitePointY
HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

Type: integer
Required: False
redPrimaryX

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

Type: integer
Required: False
Minimum: 0
Maximum: 50000

bluePrimaryX

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

Type: integer
Required: False
Minimum: 0
Maximum: 50000

bluePrimaryY

HDR Master Display Information must be provided by a color grader, using color grading tools. Range is 0 to 50,000, each increment represents 0.00002 in CIE1931 color coordinate. Note that this setting is not for color correction.

Type: integer
Required: False
Minimum: 0
Maximum: 50000

maxFrameAverageLightLevel

Maximum average light level of any frame in the coded video sequence, in units of candelas per square meter.

Type: integer
Required: True
Minimum: 0
Maximum: 65535

maxContentLightLevel

Maximum light level among all samples in the coded video sequence, in units of candelas per square meter.

Type: integer
Required: True
Minimum: 0
Maximum: 65535
minLuminance

Nominal minimum mastering display luminance in units of 0.0001 candelas per square meter

Type: integer
Required: False
Minimum: 0
Maximum: 2147483647

ImageInserter

insertableImages

Image to insert. Must be 32 bit windows BMP, PNG, or TGA file. Must not be larger than the output frames.

Type: Array of type InsertableImage (p. 828)
Required: True

InsertableImage

duration

Use Duration (Duration) to set the time, in milliseconds, for the image to remain on the output video.

Type: integer
Required: False
Minimum: -2147483648
Maximum: 2147483647

fadeOut

Use Fade out (FadeOut) to set the length, in milliseconds, of the inserted image fade out. If you don't specify a value for Fade out, the image will disappear abruptly at the end of the inserted image duration.

Type: integer
Required: False
Minimum: -2147483648
Maximum: 2147483647

imageY

Use Top (ImageY) to set the distance, in pixels, between the inserted image and the top edge of the video frame. Required for BMP, PNG and TGA input.

Type: integer
Required: True
Minimum: -2147483648
Maximum: 2147483647

fadeIn

Use Fade in (FadeIn) to set the length, in milliseconds, of the inserted image fade in. If you don't specify a value for Fade in, the image will appear abruptly at the Start time.
**imageX**

Use Left (ImageX) to set the distance, in pixels, between the inserted image and the left edge of the frame. Required for BMP, PNG and TGA input.

- **Type**: integer
- **Required**: True
- **Minimum**: -2147483648
- **Maximum**: 2147483647

**width**

Specify the Width (Width) of the inserted image. Use a value that is less than or equal to the video resolution width. Leave this setting blank to use the native width of the image.

- **Type**: integer
- **Required**: False
- **Minimum**: -2147483648
- **Maximum**: 2147483647

**startTime**

Use Start time (StartTime) to specify the video timecode when the image is inserted in the output. This must be in timecode (HH:MM:SS:FF or HH:MM:SS;FF) format.

- **Type**: string
- **Required**: False
- **Pattern**: ^(((0)?(1)|2)(0-3))((0-5)\d)(2)((0-5)\d)$

**opacity**

Use Opacity (Opacity) to specify how much of the underlying video shows through the inserted image. 0 is transparent and 100 is fully opaque. Default is 50.

- **Type**: integer
- **Required**: True
- **Minimum**: 0
- **Maximum**: 100

**layer**

Use Layer (Layer) to specify how overlapping inserted images appear. Images with higher values of layer appear on top of images with lower values of layer.

- **Type**: integer
- **Required**: True
- **Minimum**: 0
- **Maximum**: 99
**height**

Specify the Height (Height) of the inserted image. Use a value that is less than or equal to the video resolution height. Leave this setting blank to use the native height of the image.

- **Type:** integer
- **Required:** False
- **Minimum:** -2147483648
- **Maximum:** 2147483647

**imageInserterInput**

Use Image location (imageInserterInput) to specify the Amazon S3 location of the image to be inserted into the output. Use a 32 bit BMP, PNG, or TGA file that fits inside the video frame.

- **Type:** string
- **Required:** True
- **Pattern:** ^s3:\/\/(.*?)/(bmp|BMP|png|PNG|tga|TGA)$

**LanguageCode (enum)**


- ENG
- SPA
- FRA
- DEU
- GER
- ZHO
- ARA
- HIN
- JPN
- RUS
- POR
- ITA
- URD
- VIE
- KOR
- PAN
- ABK
- AAR
- AFR
- AKA
- SQI
- AMH
- ARG
- HYE
- ASM
- AWA
- AVE
- AYM
- AZE
- BAM
- BAK
<table>
<thead>
<tr>
<th>Code</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>EUS</td>
<td></td>
</tr>
<tr>
<td>BEL</td>
<td></td>
</tr>
<tr>
<td>BEN</td>
<td></td>
</tr>
<tr>
<td>BIH</td>
<td></td>
</tr>
<tr>
<td>BIS</td>
<td></td>
</tr>
<tr>
<td>BOS</td>
<td></td>
</tr>
<tr>
<td>BRE</td>
<td></td>
</tr>
<tr>
<td>BUL</td>
<td></td>
</tr>
<tr>
<td>MYA</td>
<td></td>
</tr>
<tr>
<td>CAT</td>
<td></td>
</tr>
<tr>
<td>KHM</td>
<td></td>
</tr>
<tr>
<td>CHA</td>
<td></td>
</tr>
<tr>
<td>CHE</td>
<td></td>
</tr>
<tr>
<td>NYA</td>
<td></td>
</tr>
<tr>
<td>CHU</td>
<td></td>
</tr>
<tr>
<td>CHV</td>
<td></td>
</tr>
<tr>
<td>COR</td>
<td></td>
</tr>
<tr>
<td>COS</td>
<td></td>
</tr>
<tr>
<td>CRE</td>
<td></td>
</tr>
<tr>
<td>HRV</td>
<td></td>
</tr>
<tr>
<td>CES</td>
<td></td>
</tr>
<tr>
<td>DAN</td>
<td></td>
</tr>
<tr>
<td>DIV</td>
<td></td>
</tr>
<tr>
<td>NLD</td>
<td></td>
</tr>
<tr>
<td>DZO</td>
<td></td>
</tr>
<tr>
<td>ENM</td>
<td></td>
</tr>
<tr>
<td>EPO</td>
<td></td>
</tr>
<tr>
<td>EST</td>
<td></td>
</tr>
<tr>
<td>EWE</td>
<td></td>
</tr>
<tr>
<td>FAO</td>
<td></td>
</tr>
<tr>
<td>FIJ</td>
<td></td>
</tr>
<tr>
<td>FIN</td>
<td></td>
</tr>
<tr>
<td>FRM</td>
<td></td>
</tr>
<tr>
<td>FUL</td>
<td></td>
</tr>
<tr>
<td>GLA</td>
<td></td>
</tr>
<tr>
<td>GLG</td>
<td></td>
</tr>
<tr>
<td>LUG</td>
<td></td>
</tr>
<tr>
<td>KAT</td>
<td></td>
</tr>
<tr>
<td>ELL</td>
<td></td>
</tr>
<tr>
<td>GRN</td>
<td></td>
</tr>
<tr>
<td>GUJ</td>
<td></td>
</tr>
<tr>
<td>HAT</td>
<td></td>
</tr>
<tr>
<td>HAU</td>
<td></td>
</tr>
<tr>
<td>HEB</td>
<td></td>
</tr>
<tr>
<td>HER</td>
<td></td>
</tr>
<tr>
<td>HMO</td>
<td></td>
</tr>
<tr>
<td>HUN</td>
<td></td>
</tr>
<tr>
<td>ISL</td>
<td></td>
</tr>
<tr>
<td>IDO</td>
<td></td>
</tr>
<tr>
<td>IBO</td>
<td></td>
</tr>
<tr>
<td>IND</td>
<td></td>
</tr>
<tr>
<td>INA</td>
<td></td>
</tr>
<tr>
<td>ILE</td>
<td></td>
</tr>
<tr>
<td>IKU</td>
<td></td>
</tr>
<tr>
<td>IPK</td>
<td></td>
</tr>
<tr>
<td>Language</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td></td>
</tr>
<tr>
<td>GLE</td>
<td></td>
</tr>
<tr>
<td>JAV</td>
<td></td>
</tr>
<tr>
<td>KAL</td>
<td></td>
</tr>
<tr>
<td>KAN</td>
<td></td>
</tr>
<tr>
<td>KAU</td>
<td></td>
</tr>
<tr>
<td>KAS</td>
<td></td>
</tr>
<tr>
<td>KAZ</td>
<td></td>
</tr>
<tr>
<td>KIK</td>
<td></td>
</tr>
<tr>
<td>KIN</td>
<td></td>
</tr>
<tr>
<td>KIR</td>
<td></td>
</tr>
<tr>
<td>KOM</td>
<td></td>
</tr>
<tr>
<td>KON</td>
<td></td>
</tr>
<tr>
<td>KUA</td>
<td></td>
</tr>
<tr>
<td>KUR</td>
<td></td>
</tr>
<tr>
<td>LAO</td>
<td></td>
</tr>
<tr>
<td>LAT</td>
<td></td>
</tr>
<tr>
<td>LAV</td>
<td></td>
</tr>
<tr>
<td>LIM</td>
<td></td>
</tr>
<tr>
<td>LIN</td>
<td></td>
</tr>
<tr>
<td>LIT</td>
<td></td>
</tr>
<tr>
<td>LUB</td>
<td></td>
</tr>
<tr>
<td>LTZ</td>
<td></td>
</tr>
<tr>
<td>MKD</td>
<td></td>
</tr>
<tr>
<td>MLG</td>
<td></td>
</tr>
<tr>
<td>MSA</td>
<td></td>
</tr>
<tr>
<td>MAL</td>
<td></td>
</tr>
<tr>
<td>MLT</td>
<td></td>
</tr>
<tr>
<td>GLV</td>
<td></td>
</tr>
<tr>
<td>MRI</td>
<td></td>
</tr>
<tr>
<td>MAR</td>
<td></td>
</tr>
<tr>
<td>MAH</td>
<td></td>
</tr>
<tr>
<td>MON</td>
<td></td>
</tr>
<tr>
<td>NAU</td>
<td></td>
</tr>
<tr>
<td>NAV</td>
<td></td>
</tr>
<tr>
<td>NDE</td>
<td></td>
</tr>
<tr>
<td>NBL</td>
<td></td>
</tr>
<tr>
<td>NDO</td>
<td></td>
</tr>
<tr>
<td>NEP</td>
<td></td>
</tr>
<tr>
<td>SME</td>
<td></td>
</tr>
<tr>
<td>SME</td>
<td></td>
</tr>
<tr>
<td>NOR</td>
<td></td>
</tr>
<tr>
<td>NOB</td>
<td></td>
</tr>
<tr>
<td>NNO</td>
<td></td>
</tr>
<tr>
<td>OCI</td>
<td></td>
</tr>
<tr>
<td>OJI</td>
<td></td>
</tr>
<tr>
<td>ORI</td>
<td></td>
</tr>
<tr>
<td>ORM</td>
<td></td>
</tr>
<tr>
<td>OSS</td>
<td></td>
</tr>
<tr>
<td>PLI</td>
<td></td>
</tr>
<tr>
<td>FAS</td>
<td></td>
</tr>
<tr>
<td>POL</td>
<td></td>
</tr>
<tr>
<td>PUS</td>
<td></td>
</tr>
<tr>
<td>QUE</td>
<td></td>
</tr>
<tr>
<td>QAA</td>
<td></td>
</tr>
<tr>
<td>RON</td>
<td></td>
</tr>
<tr>
<td>ROH</td>
<td></td>
</tr>
</tbody>
</table>
M2tsAudioBufferModel (enum)

Selects between the DVB and ATSC buffer models for Dolby Digital audio.
DVB
ATSC

**M2tsBufferModel (enum)**

Controls what buffer model to use for accurate interleaving. If set to MULTIPLEX, use multiplex buffer model. If set to NONE, this can lead to lower latency, but low-memory devices may not be able to play back the stream without interruptions.

- MULTIPLEX
- NONE

**M2tsEbpAudioInterval (enum)**

When set to VIDEO_AND_FIXED_INTERVALS, audio EBP markers will be added to partitions 3 and 4. The interval between these additional markers will be fixed, and will be slightly shorter than the video EBP marker interval. When set to VIDEO_INTERVAL, these additional markers will not be inserted. Only applicable when EBP segmentation markers are is selected (segmentationMarkers is EBP or EBP_LEGACY).

- VIDEO_AND_FIXED_INTERVALS
- VIDEO_INTERVAL

**M2tsEbpPlacement (enum)**

Selects which PIDs to place EBP markers on. They can either be placed only on the video PID, or on both the video PID and all audio PIDs. Only applicable when EBP segmentation markers are is selected (segmentationMarkers is EBP or EBP_LEGACY).

- VIDEO_AND_AUDIO_PIDS
- VIDEO_PID

**M2tsEsRateInPes (enum)**

Controls whether to include the ES Rate field in the PES header.

- INCLUDE
- EXCLUDE

**M2tsNielsenId3 (enum)**

If INSERT, Nielsen inaudible tones for media tracking will be detected in the input audio and an equivalent ID3 tag will be inserted in the output.

- INSERT
- NONE

**M2tsPcrControl (enum)**

When set to PCR_EVERY_PES_PACKET, a Program Clock Reference value is inserted for every Packetized Elementary Stream (PES) header. This is effective only when the PCR PID is the same as the video or audio elementary stream.

- PCR_EVERY_PES_PACKET
- CONFIGURED_PCR_PERIOD
M2tsRateMode (enum)

When set to CBR, inserts null packets into transport stream to fill specified bitrate. When set to VBR, the bitrate setting acts as the maximum bitrate, but the output will not be padded up to that bitrate.

VBR
CBR

M2tsScte35Source (enum)

Enables SCTE-35 passthrough (scte35Source) to pass any SCTE-35 signals from input to output.

PASSTHROUGH
NONE

M2tsSegmentationMarkers (enum)

 Inserts segmentation markers at each segmentation_time period. rai_segstart sets the Random Access Indicator bit in the adaptation field. rai_adapt sets the RAI bit and adds the current timecode in the private data bytes. psi_segstart inserts PAT and PMT tables at the start of segments. ebp adds Encoder Boundary Point information to the adaptation field as per OpenCable specification OC-SP-EBP-I01-130118. ebp_legacy adds Encoder Boundary Point information to the adaptation field using a legacy proprietary format.

NONE
RAI_SEGSTART
RAI_ADAPT
PSI_SEGSTART
EBP
EBP_LEGACY

M2tsSegmentationStyle (enum)

The segmentation style parameter controls how segmentation markers are inserted into the transport stream. With avails, it is possible that segments may be truncated, which can influence where future segmentation markers are inserted. When a segmentation style of "reset_cadence" is selected and a segment is truncated due to an avail, we will reset the segmentation cadence. This means the subsequent segment will have a duration of $segmentation_time seconds. When a segmentation style of "maintain_cadence" is selected and a segment is truncated due to an avail, we will not reset the segmentation cadence. This means the subsequent segment will likely be truncated as well. However, all segments after that will have a duration of $segmentation_time seconds. Note that EBP lookahead is a slight exception to this rule.

MAINTAIN_CADENCE
RESET_CADENCE

M2tsSettings

dvbTeletextPid

Packet Identifier (PID) for input source DVB Teletext data to this output.

Type: integer
Required: False
Minimum: 32
**Maximum**: 8182

**nielsenId3**
- **Type**: string
- **Required**: False

**bitrate**
The output bitrate of the transport stream in bits per second. Setting to 0 lets the muxer automatically determine the appropriate bitrate. Other common values are 3750000, 7500000, and 15000000.
- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 2147483647

**segmentationTime**
The length in seconds of each segment. Required unless markers is set to _none_.
- **Type**: number
- **Required**: False
- **Format**: float
- **Minimum**: 0.0

**audioPids**
Packet Identifier (PID) of the elementary audio stream(s) in the transport stream. Multiple values are accepted, and can be entered in ranges and/or by comma separation.
- **Type**: Array of type integer
- **Required**: False

**rateMode**
- **Type**: string
- **Required**: False

**maxPcrInterval**
Maximum time in milliseconds between Program Clock References (PCRs) inserted into the transport stream.
- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 500

**audioFramesPerPes**
The number of audio frames to insert for each PES packet.
- **Type**: integer
Properties

Required: False
Minimum: 0
Maximum: 2147483647

ebpAudioInterval

Type: string
Required: False

fragmentTime

The length in seconds of each fragment. Only used with EBP markers.

Type: number
Required: False
Format: float
Minimum: 0.0

scte35Pid

Packet Identifier (PID) of the SCTE-35 stream in the transport stream.

Type: integer
Required: False
Minimum: 32
Maximum: 8182

pmtInterval

The number of milliseconds between instances of this table in the output transport stream.

Type: integer
Required: False
Minimum: 0
Maximum: 1000

privateMetadataPid

Packet Identifier (PID) of the private metadata stream in the transport stream.

Type: integer
Required: False
Minimum: 32
Maximum: 8182

segmentationStyle

Type: string
Required: False

audioBufferModel

Type: string
Properties

ProgramNumber
The value of the program number field in the Program Map Table.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
- **Maximum:** 65535

DVB-NIT Settings
- **Type:** DvbNitSettings (p. 796)
- **Required:** False

Timed Metadata PID
Packet Identifier (PID) of the timed metadata stream in the transport stream.

- **Type:** integer
- **Required:** False
- **Minimum:** 32
- **Maximum:** 8182

SCTE-35 Source
- **Type:** string
- **Required:** False

PMT PID
Packet Identifier (PID) for the Program Map Table (PMT) in the transport stream.

- **Type:** integer
- **Required:** False
- **Minimum:** 32
- **Maximum:** 8182

Buffer Model
- **Type:** string
- **Required:** False

EBP Placement
- **Type:** string
- **Required:** False

DVB SDT Settings
- **Type:** DvbSdtSettings (p. 797)
nullPacketBitrate

Value in bits per second of extra null packets to insert into the transport stream. This can be used if a downstream encryption system requires periodic null packets.

Type: number
Required: False
Format: float
Minimum: 0.0

pcrPid

Packet Identifier (PID) of the Program Clock Reference (PCR) in the transport stream. When no value is given, the encoder will assign the same value as the Video PID.

Type: integer
Required: False
Minimum: 32
Maximum: 8182

minEbpInterval

When set, enforces that Encoder Boundary Points do not come within the specified time interval of each other by looking ahead at input video. If another EBP is going to come in within the specified time interval, the current EBP is not emitted, and the segment is "stretched" to the next marker. The lookahead value does not add latency to the system. The Live Event must be configured elsewhere to create sufficient latency to make the lookahead accurate.

Type: integer
Required: False
Minimum: 0
Maximum: 10000

transportStreamId

The value of the transport stream ID field in the Program Map Table.

Type: integer
Required: False
Minimum: 0
Maximum: 65535

videoPid

Packet Identifier (PID) of the elementary video stream in the transport stream.

Type: integer
Required: False
Minimum: 32
Maximum: 8182
pcrControl

Type: string
Required: False

esRateInPes

Type: string
Required: False

segmentationMarkers

Type: string
Required: False

dvbTdtSettings

Type: DvbTdtSettings (p. 801)
Required: False

patInterval

The number of milliseconds between instances of this table in the output transport stream.

Type: integer
Required: False
Minimum: 0
Maximum: 1000

dvbSubPids

Packet Identifier (PID) for input source DVB Subtitle data to this output. Multiple values are accepted, and can be entered in ranges and/or by comma separation.

Type: Array of type integer
Required: False

M3u8NielsenId3 (enum)

If INSERT, Nielsen inaudible tones for media tracking will be detected in the input audio and an equivalent ID3 tag will be inserted in the output.

INSERT
NONE

M3u8PcrControl (enum)

When set to PCR_EVERY_PES_PACKET a Program Clock Reference value is inserted for every Packetized Elementary Stream (PES) header. This parameter is effective only when the PCR PID is the same as the video or audio elementary stream.

PCR_EVERY_PES_PACKET
CONFIGURED_PCR_PERIOD
M3u8Scte35Source (enum)

Enables SCTE-35 passthrough (scte35Source) to pass any SCTE-35 signals from input to output.

- PASSTHROUGH
- NONE

**M3u8Settings**

**pmtPid**

Packet Identifier (PID) for the Program Map Table (PMT) in the transport stream.

- **Type**: integer
- **Required**: False
- **Minimum**: 32
- **Maximum**: 8182

**nielsenId3**

- **Type**: string
- **Required**: False

**pcrPid**

Packet Identifier (PID) of the Program Clock Reference (PCR) in the transport stream. When no value is given, the encoder will assign the same value as the Video PID.

- **Type**: integer
- **Required**: False
- **Minimum**: 32
- **Maximum**: 8182

**audioPids**

Packet Identifier (PID) of the elementary audio stream(s) in the transport stream. Multiple values are accepted, and can be entered in ranges and/or by comma separation.

- **Type**: Array of type integer
- **Required**: False

**audioFramesPerPes**

The number of audio frames to insert for each PES packet.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 2147483647

**scte35Pid**

Packet Identifier (PID) of the SCTE-35 stream in the transport stream.
Type: integer
Required: False
Minimum: 32
Maximum: 8182

**transportStreamId**

The value of the transport stream ID field in the Program Map Table.

Type: integer
Required: False
Minimum: 0
Maximum: 65535

**pcrControl**

Type: string
Required: False

**videoPid**

Packet Identifier (PID) of the elementary video stream in the transport stream.

Type: integer
Required: False
Minimum: 32
Maximum: 8182

**privateMetadataPid**

Packet Identifier (PID) of the private metadata stream in the transport stream.

Type: integer
Required: False
Minimum: 32
Maximum: 8182

**pmtInterval**

The number of milliseconds between instances of this table in the output transport stream.

Type: integer
Required: False
Minimum: 0
Maximum: 1000

**patInterval**

The number of milliseconds between instances of this table in the output transport stream.

Type: integer
Required: False
Minimum: 0
Maximum: 1000

programNumber

The value of the program number field in the Program Map Table.

Type: integer
Required: False
Minimum: 0
Maximum: 65535

timedMetadataPid

Packet Identifier (PID) of the timed metadata stream in the transport stream.

Type: integer
Required: False
Minimum: 32
Maximum: 8182

timedMetadata

Type: string
Required: False

scte35Source

Type: string
Required: False

MovClapAtom (enum)

When enabled, include 'clap' atom if appropriate for the video output settings.

   INCLUDE
   EXCLUDE

MovCslgAtom (enum)

When enabled, file composition times will start at zero, composition times in the 'ctts' (composition time to sample) box for B-frames will be negative, and a 'cslg' (composition shift least greatest) box will be included per 14496-1 amendment 1. This improves compatibility with Apple players and tools.

   INCLUDE
   EXCLUDE

MovMpeg2FourCCControl (enum)

When set to XDCAM, writes MPEG2 video streams into the QuickTime file using XDCAM fourcc codes. This increases compatibility with Apple editors and players, but may decrease compatibility with other players. Only applicable when the video codec is MPEG2.

   XDCAM
MPEG

**MovPaddingControl (enum)**

If set to OMNEON, inserts Omneon-compatible padding

- OMNEON
- NONE

**MovReference (enum)**

A value of 'external' creates separate media files and the wrapper file (.mov) contains references to these media files. A value of 'self_contained' creates only a wrapper (.mov) file and this file contains all of the media.

- SELF_CONTAINED
- EXTERNAL

**MovSettings**

- **reference**
  - Type: string
  - Required: False

- **paddingControl**
  - Type: string
  - Required: False

- **mpeg2FourCCControl**
  - Type: string
  - Required: False

- **cslgAtom**
  - Type: string
  - Required: False

- **clapAtom**
  - Type: string
  - Required: False

**Mp2Settings**

- **channels**
  
  Set Channels to specify the number of channels in this output audio track. Choosing Mono in the console will give you 1 output channel; choosing Stereo will give you 2. In the API, valid values are 1 and 2.
Properties

**Type**: integer  
**Required**: False  
**Minimum**: 1  
**Maximum**: 2

**bitrate**

Average bitrate in bits/second.

**Type**: integer  
**Required**: False  
**Minimum**: 32000  
**Maximum**: 384000

**sampleRate**

Sample rate in hz.

**Type**: integer  
**Required**: False  
**Minimum**: 32000  
**Maximum**: 48000

**Mp4CslgAtom (enum)**

When enabled, file composition times will start at zero, composition times in the 'ctts' (composition time to sample) box for B-frames will be negative, and a 'cslg' (composition shift least greatest) box will be included per 14496-1 amendment 1. This improves compatibility with Apple players and tools.

**INCLUDE**  
**EXCLUDE**

**Mp4FreeSpaceBox (enum)**

Inserts a free-space box immediately after the moov box.

**INCLUDE**  
**EXCLUDE**

**Mp4MoovPlacement (enum)**

If set to PROGRESSIVE_DOWNLOAD, the MOOV atom is relocated to the beginning of the archive as required for progressive downloading. Otherwise it is placed normally at the end.

**PROGRESSIVE_DOWNLOAD**  
**NORMAL**

**Mp4Settings**

**mp4MajorBrand**

Overrides the "Major Brand" field in the output file. Usually not necessary to specify.
Type: string
Required: False

moovPlacement
Type: string
Required: False

cslgAtom
Type: string
Required: False

freeSpaceBox
Type: string
Required: False

Mpeg2AdaptiveQuantization (enum)
Adaptive quantization. Allows intra-frame quantizers to vary to improve visual quality.

OFF
LOW
MEDIUM
HIGH

Mpeg2CodecLevel (enum)
Use Level (Mpeg2CodecLevel) to set the MPEG-2 level for the video output.

AUTO
LOW
MAIN
HIGH1440
HIGH

Mpeg2CodecProfile (enum)
Use Profile (Mpeg2CodecProfile) to set the MPEG-2 profile for the video output.

MAIN
PROFILE_422

Mpeg2FramerateControl (enum)
If you are using the console, use the Framerate setting to specify the framerate for this output. If you want to keep the same framerate as the input video, choose Follow source. If you want to do framerate conversion, choose a framerate from the dropdown list or choose Custom. The framerates shown in the dropdown list are decimal approximations of fractions. If you choose Custom, specify your framerate as a fraction. If you are creating your transcoding job specification as a JSON file without the console, use FramerateControl to specify which value the service uses for the framerate for this output. Choose INITIALIZE_FROM_SOURCE if you want the service to use the framerate from the input.
SPECIFIED if you want the service to use the framerate you specify in the settings FramerateNumerator and FramerateDenominator.

INITIALIZE_FROM_SOURCE
SPECIFIED

**Mpeg2FramerateConversionAlgorithm (enum)**

When set to INTERPOLATE, produces smoother motion during framerate conversion.

DUPLICATE_DROP
INTERPOLATE

**Mpeg2GopSizeUnits (enum)**

Indicates if the GOP Size in MPEG2 is specified in frames or seconds. If seconds the system will convert the GOP Size into a frame count at run time.

FRAMES
SECONDS

**Mpeg2InterlaceMode (enum)**

Use Interlace mode (InterlaceMode) to choose the scan line type for the output. * Top Field First (TOP_FIELD) and Bottom Field First (BOTTOM_FIELD) produce interlaced output with the entire output having the same field polarity (top or bottom first). * Follow, Default Top (FOLLOW_TOP_FIELD) and Follow, Default Bottom (FOLLOW_BOTTOM_FIELD) use the same field polarity as the source. Therefore, behavior depends on the input scan type. - If the source is interlaced, the output will be interlaced with the same polarity as the source (it will follow the source). The output could therefore be a mix of “top field first” and “bottom field first”. - If the source is progressive, the output will be interlaced with “top field first” or “bottom field first” polarity, depending on which of the Follow options you chose.

PROGRESSIVE
TOP_FIELD
BOTTOM_FIELD
FOLLOW_TOP_FIELD
FOLLOW_BOTTOM_FIELD

**Mpeg2IntraDcPrecision (enum)**

Use Intra DC precision (Mpeg2IntraDcPrecision) to set quantization precision for intra-block DC coefficients. If you choose the value auto, the service will automatically select the precision based on the per-frame compression ratio.

AUTO
INTRA_DC_PRECISION_8
INTRA_DC_PRECISION_9
INTRA_DC_PRECISION_10
INTRA_DC_PRECISION_11

**Mpeg2ParControl (enum)**

Using the API, enable ParFollowSource if you want the service to use the pixel aspect ratio from the input. Using the console, do this by choosing Follow source for Pixel aspect ratio.
INITIALIZE_FROM_SOURCE
SPECIFIED

**Mpeg2QualityTuningLevel (enum)**

Use Quality tuning level (Mpeg2QualityTuningLevel) to specify whether to use single-pass or multipass video encoding.

- SINGLE_PASS
- MULTI_PASS

**Mpeg2RateControlMode (enum)**

Use Rate control mode (Mpeg2RateControlMode) to specify whether the bitrate is variable (vbr) or constant (cbr).

- VBR
- CBR

**Mpeg2SceneChangeDetect (enum)**

Scene change detection (inserts I-frames on scene changes).

- DISABLED
- ENABLED

**Mpeg2Settings**

**minIInterval**

Enforces separation between repeated (cadence) I-frames and I-frames inserted by Scene Change Detection. If a scene change I-frame is within I-interval frames of a cadence I-frame, the GOP is shrunk and/or stretched to the scene change I-frame. GOP stretch requires enabling lookahead as well as setting I-interval. The normal cadence resumes for the next GOP. This setting is only used when Scene Change Detect is enabled. Note: Maximum GOP stretch = GOP size + Min-I-interval - 1

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 30

**parNumerator**

Pixel Aspect Ratio numerator.

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 2147483647

**gopSizeUnits**

- **Type**: string
**Properties**

**Required**: False

**hrdBufferSize**

Size of buffer (HRD buffer model) in bits. For example, enter five megabits as 5000000.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 47185920

**qualityTuningLevel**

- **Type**: string
- **Required**: False

**maxBitrate**

Maximum bitrate in bits/second. For example, enter five megabits per second as 5000000.

- **Type**: integer
- **Required**: False
- **Minimum**: 1000
- **Maximum**: 300000000

**bitrate**

Average bitrate in bits/second. Required for VBR and CBR. For MS Smooth outputs, bitrates must be unique when rounded down to the nearest multiple of 1000.

- **Type**: integer
- **Required**: False
- **Minimum**: 1000
- **Maximum**: 288000000

**spatialAdaptiveQuantization**

- **Type**: string
- **Required**: False

**slowPal**

- **Type**: string
- **Required**: False

**intraDcPrecision**

- **Type**: string
- **Required**: False

**codecProfile**

- **Type**: string
**softness**

Softness. Selects quantizer matrix, larger values reduce high-frequency content in the encoded image.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 128

**framerateControl**

- **Type**: string
- **Required**: False

**telecine**

- **Type**: string
- **Required**: False

**codecLevel**

- **Type**: string
- **Required**: False

**framerateConversionAlgorithm**

- **Type**: string
- **Required**: False

**temporalAdaptiveQuantization**

- **Type**: string
- **Required**: False

**hrdBufferInitialFillPercentage**

Percentage of the buffer that should initially be filled (HRD buffer model).

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 100

**framerateNumerator**

Framerate numerator - framerate is a fraction, e.g. 24000 / 1001 = 23.976 fps.

- **Type**: integer
- **Required**: False
- **Minimum**: False
- **Maximum**: 24
Maximum: 60000

gopClosedCadence
Frequency of closed GOPs. In streaming applications, it is recommended that this be set to 1 so a decoder joining mid-stream will receive an IDR frame as quickly as possible. Setting this value to 0 will break output segmenting.

Type: integer
Required: False
Minimum: 0
Maximum: 2147483647

numberBFramesBetweenReferenceFrames
Number of B-frames between reference frames.

Type: integer
Required: False
Minimum: 0
Maximum: 7

framerateDenominator
Framerate denominator.

Type: integer
Required: False
Minimum: 1
Maximum: 1001

adaptiveQuantization

Type: string
Required: False

interlaceMode

Type: string
Required: False

gopSize
GOP Length (keyframe interval) in frames or seconds. Must be greater than zero.

Type: number
Required: False
Format: float
Minimum: 0.0

syntax

Type: string
Required: False

**parDenominator**

Pixel Aspect Ratio denominator.

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 2147483647

**sceneChangeDetect**

- **Type**: string
- **Required**: False

**parControl**

- **Type**: string
- **Required**: False

**rateControlMode**

- **Type**: string
- **Required**: False

**Mpeg2SlowPal (enum)**

Enables Slow PAL rate conversion. 23.976fps and 24fps input is relabeled as 25fps, and audio is sped up correspondingly.

- **DISABLED**
- **ENABLED**

**Mpeg2SpatialAdaptiveQuantization (enum)**

Adjust quantization within each frame based on spatial variation of content complexity.

- **DISABLED**
- **ENABLED**

**Mpeg2Syntax (enum)**

Produces a Type D-10 compatible bitstream (SMPTE 356M-2001).

- **DEFAULT**
- **D_10**

**Mpeg2Telecine (enum)**

Only use Telecine (Mpeg2Telecine) when you set Framerate (Framerate) to 29.970. Set Telecine (Mpeg2Telecine) to Hard (hard) to produce a 29.97i output from a 23.976 input. Set it to Soft (soft) to produce 23.976 output and leave conversion to the player.
NONE
SOFT
HARD

Mpeg2TemporalAdaptiveQuantization (enum)

Adjust quantization within each frame based on temporal variation of content complexity.

DISABLED
ENABLED

NoiseReducer

filter

Type: string
Required: True

filterSettings

Type: NoiseReducerFilterSettings (p. 853)
Required: False

spatialFilterSettings

Type: NoiseReducerSpatialFilterSettings (p. 854)
Required: False

NoiseReducerFilter (enum)

Use Noise reducer filter (NoiseReducerFilter) to select one of the following spatial image filtering functions. To use this setting, you must also enable Noise reducer (NoiseReducer). * Bilateral is an edge preserving noise reduction filter. * Mean (softest), Gaussian, Lanczos, and Sharpen (sharpest) are convolution filters. * Conserve is a min/max noise reduction filter. * Spatial is a frequency-domain filter based on JND principles.

BILATERAL
MEAN
GAUSSIAN
LANCZOS
SHARPEN
CONSERVE
SPATIAL

NoiseReducerFilterSettings

strength

Relative strength of noise reducing filter. Higher values produce stronger filtering.

Type: integer
Required: False
Properties

Minimum: 0
Maximum: 3

**NoiseReducerSpatialFilterSettings**

**strength**

Relative strength of noise reducing filter. Higher values produce stronger filtering.

- Type: integer
- Required: False
- Minimum: 0
- Maximum: 16

**postFilterSharpenStrength**

Specify strength of post noise reduction sharpening filter, with 0 disabling the filter and 3 enabling it at maximum strength.

- Type: integer
- Required: False
- Minimum: 0
- Maximum: 3

**speed**

The speed of the filter, from -2 (lower speed) to 3 (higher speed), with 0 being the nominal value.

- Type: integer
- Required: False
- Minimum: -2
- Maximum: 3

**OutputChannelMapping**

**inputChannels**

List of input channels

- Type: Array of type integer
- Required: True

**OutputSdt (enum)**

Selects method of inserting SDT information into output stream. "Follow input SDT" copies SDT information from input stream to output stream. "Follow input SDT if present" copies SDT information from input stream to output stream if SDT information is present in the input, otherwise it will fall back on the user-defined values. Enter "SDT Manually" means user will enter the SDT information. "No SDT" means output stream will not contain SDT information.

- SDT_FOLLOW
- SDT_FOLLOW_IF_PRESENT
- SDT_MANUAL
Preset

createdAt
The timestamp in epoch seconds for preset creation.

Type: string
Required: False
Format: date-time

lastUpdated
The timestamp in epoch seconds when the preset was last updated.

Type: string
Required: False
Format: date-time

settings
Type: PresetSettings (p. 856)
Required: True

name
A name you create for each preset. Each name must be unique within your account.

Type: string
Required: True

description
An optional description you create for each preset.

Type: string
Required: False

arn
An identifier for this resource that is unique within all of AWS.

Type: string
Required: False

category
An optional category you create to organize your presets.

Type: string
Required: False
type
A preset can be of two types: system or custom. System or built-in preset can't be modified or deleted by
the user.

  Type: string
  Required: False

PresetSettings

videoDescription
(VideoDescription) contains a group of video encoding settings. The specific video settings depend on
the video codec you choose when you specify a value for Video codec (codec). Include one instance of
(VideoDescription) per output.

  Type: VideoDescription (p. 864)
  Required: False

audioDescriptions
(AudioDescriptions) contains groups of audio encoding settings organized by audio codec. Include one
instance of (AudioDescriptions) per output. (AudioDescriptions) can contain multiple groups of encoding
settings.

  Type: Array of type AudioDescription (p. 784)
  Required: False

containerSettings

  Type: ContainerSettings (p. 794)
  Required: False

captionDescriptions
Caption settings for this preset. There can be multiple caption settings in a single output.

  Type: Array of type CaptionDescriptionPreset (p. 791)
  Required: False

ProresCodecProfile (enum)
Use Profile (ProResCodecProfile) to specify the type of Apple ProRes codec to use for this output.

  APPLE_PRORES_422
  APPLE_PRORES_422_HQ
  APPLE_PRORES_422_LT
  APPLE_PRORES_422_PROXY

ProresFramerateControl (enum)
If you are using the console, use the Framerate setting to specify the framerate for this output. If
you want to keep the same framerate as the input video, choose Follow source. If you want to do
framerate conversion, choose a framerate from the dropdown list or choose Custom. The framerates shown in the dropdown list are decimal approximations of fractions. If you choose Custom, specify your framerate as a fraction. If you are creating your transcoding job specification as a JSON file without the console, use FramerateControl to specify which value the service uses for the framerate for this output. Choose INITIALIZE_FROM_SOURCE if you want the service to use the framerate from the input. Choose SPECIFIED if you want the service to use the framerate you specify in the settings FramerateNumerator and FramerateDenominator.

```
INITIALIZE_FROM_SOURCE
SPECIFIED
```

**ProresFramerateConversionAlgorithm (enum)**

When set to INTERPOLATE, produces smoother motion during framerate conversion.

```
DUPLICATE_DROP
INTERPOLATE
```

**ProresInterlaceMode (enum)**

Use Interlace mode (InterlaceMode) to choose the scan line type for the output. * Top Field First (TOP_FIELD) and Bottom Field First (BOTTOM_FIELD) produce interlaced output with the entire output having the same field polarity (top or bottom first). * Follow, Default Top (FOLLOW_TOP_FIELD) and Follow, Default Bottom (FOLLOW_BOTTOM_FIELD) use the same field polarity as the source. Therefore, behavior depends on the input scan type. - If the source is interlaced, the output will be interlaced with the same polarity as the source (it will follow the source). The output could therefore be a mix of "top field first" and "bottom field first". - If the source is progressive, the output will be interlaced with "top field first" or "bottom field first" polarity, depending on which of the Follow options you chose.

```
PROGRESSIVE
TOP_FIELD
BOTTOM_FIELD
FOLLOW_TOP_FIELD
FOLLOW_BOTTOM_FIELD
```

**ProresParControl (enum)**

Use (ProresParControl) to specify how the service determines the pixel aspect ratio. Set to Follow source (INITIALIZE_FROM_SOURCE) to use the pixel aspect ratio from the input. To specify a different pixel aspect ratio: Using the console, choose it from the dropdown menu. Using the API, set ProresParControl to (SPECIFIED) and provide for (ParNumerator) and (ParDenominator).

```
INITIALIZE_FROM_SOURCE
SPECIFIED
```

**ProresSettings**

**framerateDenominator**

Framerate denominator.

```
Type: integer
Required: False
Minimum: 1
Maximum: 2147483647
```
slowPal
Type: string
Required: False

framerateControl
Type: string
Required: False

telecine
Type: string
Required: False

framerateConversionAlgorithm
Type: string
Required: False

interlaceMode
Type: string
Required: False

parNumerator
Pixel Aspect Ratio numerator.
Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

codecProfile
Type: string
Required: False

parDenominator
Pixel Aspect Ratio denominator.
Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

parControl
Type: string
Required: False
framerateNumerator

When you use the API for transcode jobs that use framerate conversion, specify the framerate as a fraction. For example, \( \frac{24000}{1001} = 23.976 \) fps. Use FramerateNumerator to specify the numerator of this fraction. In this example, use 24000 for the value of FramerateNumerator.

Type: integer
Required: False
Minimum: 1
Maximum: 2147483647

ProresSlowPal (enum)

Enables Slow PAL rate conversion. 23.976fps and 24fps input is relabeled as 25fps, and audio is sped up correspondingly.

DISABLED
ENABLED

ProresTelecine (enum)

Only use Telecine (ProresTelecine) when you set Framerate (Framerate) to 29.970. Set Telecine (ProresTelecine) to Hard (hard) to produce a 29.97i output from a 23.976 input. Set it to Soft (soft) to produce 23.976 output and leave conversion to the player.

NONE
HARD

Rectangle

width

Width of rectangle in pixels.

Type: integer
Required: True
Minimum: -2147483648
Maximum: 2147483647

x

The distance, in pixels, between the rectangle and the left edge of the video frame.

Type: integer
Required: True
Minimum: -2147483648
Maximum: 2147483647

y

The distance, in pixels, between the rectangle and the top edge of the video frame.

Type: integer
Required: True
Minimum: -2147483648
Maximum: 2147483647
height

Height of rectangle in pixels.

Type: integer
Required: True
Minimum: -2147483648
Maximum: 2147483647

RemixSettings

channelsOut

Specify the number of channels in this output after remixing. Valid values: 1, 2, 4, 6, 8

Type: integer
Required: True
Minimum: 1
Maximum: 8

channelMapping

Type: ChannelMapping (p. 793)
Required: True

channelsIn

Specify the number of audio channels from your input that you want to use in your output. With remixing, you might combine or split the data in these channels, so the number of channels in your final output might be different.

Type: integer
Required: True
Minimum: 1
Maximum: 16

RespondToAfd (enum)

Use Respond to AFD (RespondToAfd) to specify how the service changes the video itself in response to AFD values in the input. * Choose Respond to clip the input video frame according to the AFD value, input display aspect ratio, and output display aspect ratio. * Choose Passthrough to include the input AFD values. Do not choose this when AfdSignaling is set to (NONE). A preferred implementation of this workflow is to set RespondToAfd to (NONE) and set AfdSignaling to (AUTO). * Choose None to remove all input AFD values from this output.

NONE
RESPOND
PASSTHROUGH

ScalingBehavior (enum)

Applies only if your input aspect ratio is different from your output aspect ratio. Enable Stretch to output (StretchToOutput) to have the service stretch your video image to fit. Leave this setting disabled to
allow the service to letterbox your video instead. This setting overrides any positioning value you specify elsewhere in the job.

DEFAULT
STRETCH_TO_OUTPUT

SccDestinationFramerate (enum)

Set Framerate (SccDestinationFramerate) to make sure that the captions and the video are synchronized in the output. Specify a framerate that matches the framerate of the associated video. If the video framerate is 29.97, choose 29.97 dropframe (FRAMERATE_29_97_DROPFRAME) only if the video has video_insertion=true and drop_frame_timecode=true; otherwise, choose 29.97 non-dropframe (FRAMERATE_29_97_NON_DROPFRAME).

FRAMERATE_23_97
FRAMERATE_24
FRAMERATE_29_97_DROPFRAME
FRAMERATE_29_97_NON_DROPFRAME

SccDestinationSettings

framerate

Type: string
Required: False

TeletextDestinationSettings

pageNumber

Set pageNumber to the Teletext page number for the destination captions for this output. This value must be a three-digit hexadecimal string; strings ending in -FF are invalid. If you are passing through the entire set of Teletext data, do not use this field.

Type: string
Required: False
Pattern: \^[1-8][0-9a-fA-F][0-9a-eA-E]$

TimecodeBurnin

prefix

Use Prefix (Prefix) to place ASCII characters before any burned-in timecode. For example, a prefix of "EZ-" will result in the timecode "EZ-00:00:00:00". Provide either the characters themselves or the ASCII code equivalents. The supported range of characters is 0x20 through 0x7e. This includes letters, numbers, and all special characters represented on a standard English keyboard.

Type: string
Required: False
Pattern: \^[ -~]+$

fontSize

Use Font Size (FontSize) to set the font size of any burned-in timecode. Valid values are 10, 16, 32, 48.
Properties

Type: integer
Required: False
Minimum: 10
Maximum: 48

position

Type: string
Required: False

TimecodeBurninPosition (enum)

Use Position (Position) under Timecode burn-in (TimecodeBurnIn) to specify the location the burned-in timecode on output video.

- TOP_CENTER
- TOP_LEFT
- TOP_RIGHT
- MIDDLE_LEFT
- MIDDLE_CENTER
- MIDDLE_RIGHT
- BOTTOM_LEFT
- BOTTOM_CENTER
- BOTTOM_RIGHT

TimedMetadata (enum)

Applies only to HLS outputs. Use this setting to specify whether the service inserts the ID3 timed metadata from the input in this output.

- PASSTHROUGH
- NONE

TtmlDestinationSettings

stylePassthrough

Type: string
Required: False

TtmlStylePassthrough (enum)

Pass through style and position information from a TTML-like input source (TTML, SMPTE-TT, CFF-TT) to the CFF-TT output or TTML output.

- ENABLED
- DISABLED

Type (enum)

- SYSTEM
- CUSTOM
UpdatePresetRequest

settings

  Type: PresetSettings (p. 856)
  Required: False

name

The name of the preset you are modifying.

  Type: string
  Required: False

description

The new description for the preset, if you are changing it.

  Type: string
  Required: False

category

The new category for the preset, if you are changing it.

  Type: string
  Required: False

UpdatePresetResponse

preset

  Type: Preset (p. 855)
  Required: False

VideoCodec (enum)

Type of video codec

  FRAME_CAPTURE
  H_264
  H_265
  MPEG2
  PRORES

VideoCodecSettings

h265Settings

  Type: H265Settings (p. 819)
  Required: False
Properties

codec
Type: string
Required: True

proresSettings
Type: ProresSettings (p. 857)
Required: False

mpeg2Settings
Type: Mpeg2Settings (p. 848)
Required: False

frameCaptureSettings
Type: FrameCaptureSettings (p. 807)
Required: False

h264Settings
Type: H264Settings (p. 811)
Required: False

VideoDescription

fixedAfd
Applies only if you set AFD Signaling(AfdSignaling) to Fixed (FIXED). Use Fixed (FixedAfd) to specify a four-bit AFD value which the service will write on all frames of this video output.

Type: integer
Required: False
Minimum: 0
Maximum: 15

scalingBehavior
Type: string
Required: False

respondToAfd
Type: string
Required: False

codecSettings
Type: VideoCodecSettings (p. 863)
Required: True
**afdSignaling**

- **Type:** string
- **Required:** False

**colorMetadata**

- **Type:** string
- **Required:** False

**timecodeInsertion**

- **Type:** string
- **Required:** False

**width**

Use **Width (Width)** to define the video resolution width, in pixels, for this output. If you don’t provide a value here, the service will use the input width.

- **Type:** integer
- **Required:** False
- **Minimum:** 32
- **Maximum:** 4096

**videoPreprocessors**

Find additional transcoding features under **Preprocessors (VideoPreprocessors)**. Enable the features at each output individually. These features are disabled by default.

- **Type:** VideoPreprocessor (p. 866)
- **Required:** False

**antiAlias**

- **Type:** string
- **Required:** False

**position**

Use **Position (Position)** to point to a rectangle object to define your position. This setting overrides any other aspect ratio.

- **Type:** Rectangle (p. 859)
- **Required:** False

**sharpness**

Use **Sharpness (Sharpness)** setting to specify the strength of anti-aliasing. This setting changes the width of the anti-alias filter kernel used for scaling. Sharpness only applies if your output resolution is different from your input resolution, and if you set Anti-alias (AntiAlias) to ENABLED. 0 is the softest setting, 100 the sharpest, and 50 recommended for most content.

- **Type:** integer
Properties

 Required: False  
 Minimum: 0  
 Maximum: 100

crop

Applies only if your input aspect ratio is different from your output aspect ratio. Use Input cropping rectangle (Crop) to specify the video area the service will include in the output. This will crop the input source, causing video pixels to be removed on encode. Do not use this setting if you have enabled Stretch to output (stretchToOutput) in your output settings.

 Type: Rectangle (p. 859)  
 Required: False

dropFrameTimecode

 Type: string  
 Required: False

height

Use the Height (Height) setting to define the video resolution height for this output. Specify in pixels. If you don’t provide a value here, the service will use the input height.

 Type: integer  
 Required: False  
 Minimum: 32  
 Maximum: 2160

VideoPreprocessor

noiseReducer

Enable the Noise reducer (NoiseReducer) feature to remove noise from your video output if necessary. Enable or disable this feature for each output individually. This setting is disabled by default.

 Type: NoiseReducer (p. 853)  
 Required: False

timecodeBurnin

Timecode burn-in (TimecodeBurnIn)--Burns the output timecode and specified prefix into the output.

 Type: TimecodeBurnin (p. 861)  
 Required: False

colorCorrector

Enable the Color corrector (ColorCorrector) feature if necessary. Enable or disable this feature for each output individually. This setting is disabled by default.

 Type: ColorCorrector (p. 793)  
 Required: False
**deinterlacer**

Use Deinterlacer (Deinterlacer) to produce smoother motion and a clearer picture.

**Type**: Deinterlacer (p. 795)

**Required**: False

**imageInserter**

Enable the Image inserter (ImageInserter) feature to include a graphic overlay on your video. Enable or disable this feature for each output individually. This setting is disabled by default.

**Type**: ImageInserter (p. 828)

**Required**: False

**VideoTimecodeInsertion (enum)**

Applies only to H.264, H.265, MPEG2, and ProRes outputs. Only enable Timecode insertion when the input framerate is identical to the output framerate. To include timecodes in this output, set Timecode insertion (VideoTimecodeInsertion) to PIC_TIMING_SEI. To leave them out, set it to DISABLED. Default is DISABLED. When the service inserts timecodes in an output, by default, it uses any embedded timecodes from the input. If none are present, the service will set the timecode for the first output frame to zero. To change this default behavior, adjust the settings under Timecode configuration (TimecodeConfig). In the console, these settings are located under Job > Job settings > Timecode configuration. Note - Timecode source under input settings (InputTimecodeSource) does not affect the timecodes that are inserted in the output. Source under Job settings > Timecode configuration (TimecodeSource) does.

- DISABLED
- PIC_TIMING_SEI

**WavFormat (enum)**

The service defaults to using RIFF for WAV outputs. If your output audio is likely to exceed 4 GB in file size, or if you otherwise need the extended support of the RF64 format, set your output WAV file format to RF64.

- RIFF
- RF64

**WavSettings**

**channels**

Set Channels to specify the number of channels in this output audio track. With WAV, valid values 1, 2, 4, and 8. In the console, these values are Mono, Stereo, 4-Channel, and 8-Channel, respectively.

**Type**: integer

**Required**: False

**Minimum**: 1

**Maximum**: 8

**bitDepth**

Specify Bit depth (BitDepth), in bits per sample, to choose the encoding quality for this audio track.
Queues

URI

/2017-08-29/queues

HTTP Methods

GET

Operation ID: ListQueues

Retrieve a JSON array of up to twenty of your queues. This will return the queues themselves, not just a list of them. To retrieve the next twenty queues, use the nextToken string returned with the array.

Query Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>listBy</td>
<td>String</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>nextToken</td>
<td>String</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>maxResults</td>
<td>String</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>order</td>
<td>String</td>
<td>False</td>
<td>False</td>
</tr>
</tbody>
</table>

Responses

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>ListQueuesResponse (p. 870)</td>
<td>200 response</td>
</tr>
</tbody>
</table>
POST

Operation ID: CreateQueue

Create a new transcoding queue. For information about job templates see the User Guide at http://docs.aws.amazon.com/mediaconvert/latest/ug/what-is.html

Responses

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>CreateQueueResponse (p. 871)</td>
<td>201 response</td>
</tr>
<tr>
<td>400</td>
<td>ExceptionBody (p. 870)</td>
<td>BadRequestException</td>
</tr>
<tr>
<td>500</td>
<td>ExceptionBody (p. 870)</td>
<td>InternalServiceException</td>
</tr>
<tr>
<td>403</td>
<td>ExceptionBody (p. 870)</td>
<td>AccessDeniedException</td>
</tr>
<tr>
<td>404</td>
<td>ExceptionBody (p. 870)</td>
<td>ResourceNotFoundException</td>
</tr>
<tr>
<td>429</td>
<td>ExceptionBody (p. 870)</td>
<td>LimitExceededException</td>
</tr>
<tr>
<td>409</td>
<td>ExceptionBody (p. 870)</td>
<td>ResourceInUseException</td>
</tr>
</tbody>
</table>

Schemas

Request Bodies

Example GET

```json
{
  "nextToken (p. 871)": "string",
  "maxResults (p. 871)": integer,
  "listBy (p. 871)": enum,
  "order (p. 871)": enum
}
```

Example POST

```json
{
```
### Properties

#### CreateQueueRequest

**name**

The name of the queue you are creating.

*Type: string*
Required: True

description
Optional. A description of the queue you are creating.
  Type: string
  Required: False

CreateQueueResponse
queue
  Type: Queue (p. 872)
  Required: False

ExceptionBody
message
  Type: string
  Required: False

ListQueuesRequest
nextToken
Use this string, provided with the response to a previous request, to request the next batch of queues.
  Type: string
  Required: False

maxResults
Optional. Number of queues, up to twenty, that will be returned at one time.
  Type: integer
  Required: False
  Format: int32
  Minimum: 1
  Maximum: 20

listBy
  Type: string
  Required: False

order
  Type: string
  Required: False
**ListQueuesResponse**

queues

List of queues

  * **Type:** Array of type Queue (p. 872)
  * **Required:** False

nextToken

Use this string to request the next batch of queues.

  * **Type:** string
  * **Required:** False

**Order (enum)**

When you request lists of resources, you can optionally specify whether they are sorted in ASCENDING or DESCENDING order. Default varies by resource.

  * ASCENDING
  * DESCENDING

**Queue**

createdAt

The timestamp in epoch seconds for queue creation.

  * **Type:** string
  * **Required:** False
  * **Format:** date-time

lastUpdated

The timestamp in epoch seconds when the queue was last updated.

  * **Type:** string
  * **Required:** False
  * **Format:** date-time

progressingJobsCount

Estimated number of jobs in PROGRESSING status.

  * **Type:** integer
  * **Required:** False
  * **Format:** int64

submittedJobsCount

Estimated number of jobs in SUBMITTED status.
Properties

Type: integer
Required: False
Format: int64

name

A name you create for each queue. Each name must be unique within your account.

Type: string
Required: True

description

An optional description you create for each queue.

Type: string
Required: False

arn

An identifier for this resource that is unique within all of AWS.

Type: string
Required: False

type

A queue can be of two types: system or custom. System or built-in queues can't be modified or deleted by the user.

Type: string
Required: False

status

Type: string
Required: False

QueueListBy (enum)

Optional. When you request a list of queues, you can choose to list them alphabetically by NAME or chronologically by CREATION_DATE. If you don't specify, the service will list them by creation date.

NAME
CREATION_DATE

QueueStatus (enum)

Queues can be ACTIVE or PAUSED. If you pause a queue, jobs in that queue will not begin. Jobs running when a queue is paused continue to run until they finish or error out.

ACTIVE
PAUSED
Type (enum)

SYSTEM
CUSTOM

_queues name_{

**URI**

/2017-08-29/queues/ _name_

**HTTP Methods**

**GET**

Operation ID: GetQueue

Retrieve the JSON for a specific queue.

Path Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>True</td>
<td></td>
</tr>
</tbody>
</table>

Responses

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>GetQueueResponse (p. 876)</td>
<td>200 response</td>
</tr>
<tr>
<td>400</td>
<td>ExceptionBody (p. 876)</td>
<td>BadRequestException</td>
</tr>
<tr>
<td>500</td>
<td>ExceptionBody (p. 876)</td>
<td>InternalServiceException</td>
</tr>
<tr>
<td>403</td>
<td>ExceptionBody (p. 876)</td>
<td>AccessDeniedException</td>
</tr>
<tr>
<td>404</td>
<td>ExceptionBody (p. 876)</td>
<td>ResourceNotFoundException</td>
</tr>
<tr>
<td>429</td>
<td>ExceptionBody (p. 876)</td>
<td>LimitExceededException</td>
</tr>
<tr>
<td>409</td>
<td>ExceptionBody (p. 876)</td>
<td>ResourceInUseException</td>
</tr>
</tbody>
</table>

**PUT**

Operation ID: UpdateQueue

Modify one of your existing queues.

Path Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>True</td>
<td></td>
</tr>
</tbody>
</table>
DELETE

Operation ID: DeleteQueue

Permanently delete a queue you have created.

Path Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>True</td>
<td></td>
</tr>
</tbody>
</table>

Responses

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>ExceptionBody (p. 876)</td>
<td>BadRequestException</td>
</tr>
<tr>
<td>202</td>
<td>DeleteQueueResponse (p. 876)</td>
<td>202 response</td>
</tr>
<tr>
<td>500</td>
<td>ExceptionBody (p. 876)</td>
<td>InternalServiceException</td>
</tr>
<tr>
<td>403</td>
<td>ExceptionBody (p. 876)</td>
<td>AccessDeniedException</td>
</tr>
<tr>
<td>404</td>
<td>ExceptionBody (p. 876)</td>
<td>ResourceNotFoundException</td>
</tr>
<tr>
<td>429</td>
<td>ExceptionBody (p. 876)</td>
<td>LimitExceededException</td>
</tr>
<tr>
<td>409</td>
<td>ExceptionBody (p. 876)</td>
<td>ResourceInUseException</td>
</tr>
</tbody>
</table>

Schemas

Request Bodies

Example GET

```json
{
  "name (p. 877)" : "string"
}
```
Example PUT

```json
{
  "name (p. 879)": "string",
  "description (p. 879)": "string",
  "status (p. 879)": enum
}
```

Example DELETE

```json
{
  "name (p. 877)": "string"
}
```

Response Bodies

Example GetQueueResponse

```json
{
  "queue (p. 877)": {
    "createdAt (p. 877)": "string",
    "lastUpdated (p. 877)": "string",
    "progressingJobsCount (p. 878)": integer,
    "submittedJobsCount (p. 878)": integer,
    "name (p. 878)": "string",
    "description (p. 878)": "string",
    "arn (p. 878)": "string",
    "type (p. 878)": enum,
    "status (p. 878)": enum
  }
}
```

Example UpdateQueueResponse

```json
{
  "queue (p. 879)": {
    "createdAt (p. 877)": "string",
    "lastUpdated (p. 877)": "string",
    "progressingJobsCount (p. 878)": integer,
    "submittedJobsCount (p. 878)": integer,
    "name (p. 878)": "string",
    "description (p. 878)": "string",
    "arn (p. 878)": "string",
    "type (p. 878)": enum,
    "status (p. 878)": enum
  }
}
```

Example DeleteQueueResponse

```json
{
}
```

Example ExceptionBody

```json
{
  "message (p. 877)": "string"
}
```
Properties

DeleteQueueRequest

name
The name of the queue to be deleted.

Type: string
Required: False

DeleteQueueResponse

ExceptionBody

message

Type: string
Required: False

GetQueueRequest

name
The name of the queue.

Type: string
Required: False

GetQueueResponse

queue

Type: Queue (p. 877)
Required: False

Queue

createdAt
The timestamp in epoch seconds for queue creation.

Type: string
Required: False
Format: date-time

lastUpdated
The timestamp in epoch seconds when the queue was last updated.
Type: string
Required: False
Format: date-time

**progressingJobsCount**

Estimated number of jobs in PROGRESSING status.

Type: integer
Required: False
Format: int64

**submittedJobsCount**

Estimated number of jobs in SUBMITTED status.

Type: integer
Required: False
Format: int64

**name**

A name you create for each queue. Each name must be unique within your account.

Type: string
Required: True

**description**

An optional description you create for each queue.

Type: string
Required: False

**arn**

An identifier for this resource that is unique within all of AWS.

Type: string
Required: False

**type**

A queue can be of two types: system or custom. System or built-in queues can't be modified or deleted by the user.

Type: string
Required: False

**status**

Type: string
Required: False
QueueStatus (enum)

Queues can be ACTIVE or PAUSED. If you pause a queue, jobs in that queue will not begin. Jobs running when a queue is paused continue to run until they finish or error out.

- ACTIVE
- PAUSED

Type (enum)

- SYSTEM
- CUSTOM

UpdateQueueRequest

name

The name of the queue you are modifying.

- Type: string
- Required: False

description

The new description for the queue, if you are changing it.

- Type: string
- Required: False

status

- Type: string
- Required: False

UpdateQueueResponse

queue

- Type: Queue (p. 877)
- Required: False
# Document History

The following table describes important changes to this documentation.

- **API version:** latest
- **Latest documentation update:** June 12, 2018

<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>New CMAF output group</td>
<td>cmafGroupSettings and its children added to schema, under OutputGroupSettings.</td>
<td>June 12, 2018</td>
</tr>
<tr>
<td>New Getting Started Using SDKs or CLI</td>
<td>Added chapter that shows how to get your custom endpoint and send AWS Elemental MediaConvert requests to it. Includes examples in various programming languages.</td>
<td>May 17, 2018</td>
</tr>
<tr>
<td>New AWS Elemental MediaConvert service release</td>
<td>Initial documentation for the AWS Elemental MediaConvert service.</td>
<td>November 27, 2017</td>
</tr>
</tbody>
</table>
AWS Glossary

For the latest AWS terminology, see the AWS Glossary in the AWS General Reference.