AWS Elemental
MediaLive API Reference

API Reference
AWS Elemental MediaLive API Reference: API Reference
Copyright © 2019 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not
Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or
discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may
or may not be affiliated with, connected to, or sponsored by Amazon.
What Is the AWS Elemental MediaLive API?

This is the AWS Elemental MediaLive REST API Reference. It provides information on the URL, request contents, and response contents of each AWS Elemental MediaLive REST operation.

We assume that your IAM user credentials have the permissions needed to use AWS Elemental MediaLive via the REST API. We also assume that you are familiar with the features and operations of AWS Elemental MediaLive, as described in the user guide.

For general information on the service, see the AWS Elemental MediaLive User Guide.
Resources

The AWS Elemental MediaLive REST API includes the following resources.

Topics

- Channels (p. 2)
- Channels channelId (p. 140)
- Channels channelId ChannelClass (p. 285)
- Channels channelId Schedule (p. 407)
- Channels channelId Start (p. 431)
- Channels channelId Stop (p. 551)
- InputSecurityGroups (p. 671)
- InputSecurityGroups inputSecurityGroupId (p. 677)
- Inputs (p. 684)
- Inputs inputId (p. 696)
- Offerings (p. 708)
- Offerings offeringId (p. 716)
- Offerings offeringId Purchase (p. 723)
- Reservations (p. 733)
- Reservations reservationId (p. 743)
- Tags resource-arn (p. 754)

Channels

URI

/prod/channels

HTTP Methods

GET

Operation ID: ListChannels

Produces list of channels that have been created

Query Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>

Responses

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>ListChannelsResultModel</td>
<td>array of channels</td>
</tr>
</tbody>
</table>
### Status Code Response Model Description

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>InvalidRequest (p. 28)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 28)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 29)</td>
<td>Request limit exceeded on list channel calls to channel service.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 29)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>BadGatewayException (p. 29)</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>504</td>
<td>GatewayTimeoutException (p. 29)</td>
<td>Gateway Timeout Error</td>
</tr>
</tbody>
</table>

### POST

**Operation ID: CreateChannel**

Creates a new channel

**Responses**

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>CreateChannelResultModel (p. 17)</td>
<td>Creation of channel is started.</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 28)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 28)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>409</td>
<td>ResourceConflict (p. 28)</td>
<td>The channel is unable to create due to an issue with channel resources.</td>
</tr>
<tr>
<td>422</td>
<td>ChannelConfigurationValidationError (p. 28)</td>
<td>The channel failed validation and could not be created.</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 29)</td>
<td>Request limit exceeded on list channel calls to channel service.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 29)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>BadGatewayException (p. 29)</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>504</td>
<td>GatewayTimeoutException (p. 29)</td>
<td>Gateway Timeout Error</td>
</tr>
</tbody>
</table>

### Schemas

**Request Bodies**

**Example POST**

```
{
```
"inputAttachments": [  
  
  
  
  
  "inputId": "string",
  "inputAttachmentName": "string",
  "inputSettings": { 
    "sourceEndBehavior": enum,
    "deblockFilter": enum,
    "audioSelectors": [ 
      
      
      
      
      "name": "string",
      "selectorSettings": { 
        "audioLanguageSelection": { 
          "languageSelectionPolicy": enum,
          "languageCode": "string"
        },
        "audioPidSelection": { 
          "pid": integer
        }
      }
    },
    "networkInputSettings": { 
      "hlsInputSettings": { 
        "retries": integer,
        "bandwidth": integer,
        "retryInterval": integer,
        "bufferSegments": integer
      },
      "serverValidation": enum
    },
    "inputFilter": enum,
    "videoSelector": { 
      "colorSpace": enum,
      "selectorSettings": { 
        "videoSelectorPid": { 
          "pid": integer
        },
        "videoSelectorProgramId": { 
          "programId": integer
        }
      },
      "colorSpaceUsage": enum
    },
    "filterStrength": integer,
    "denoiseFilter": enum,
    "captionSelectors": [ 
      
      
      "name": "string",
      "languageCode": "string",
      "selectorSettings": { 
        "embeddedSourceSettings": { 
          "scte20Detection": enum,
          "source608ChannelNumber": integer,
          "convert608To708": enum,
          "source608TrackNumber": integer
        },
        "scte35SourceSettings": { 
          "source608ChannelNumber": integer,
          "convert608To708": enum
        },
        "dvbSubSourceSettings": { 
          "pid": integer
        },
        "teletextSourceSettings": { 
          "pageNumber": "string"
        }
      }
    ]
  }
}
"aribSourceSettings": {
},
"scte27SourceSettings": {
"pid": integer
}
},
"logLevel": enum,
"reserved": "string",
"requestId": "string",
"roleArn": "string",
"destinations": [
{
"mediaPackageSettings": [
{
"channelId": "string"
}
],
"settings": [
{
"passwordParam": "string",
"streamName": "string",
"url": "string",
"username": "string"
}
],
"id": "string"
}
],
"name": "string",
"encoderSettings": {
"timecodeConfig": {
"syncThreshold": integer,
"source": enum
},
"outputGroups": [
{
"outputs": [
{
"videoDescriptionName": "string",
"outputName": "string",
"captionDescriptionNames": ["string"
],
"outputSettings": {
"rtmpOutputSettings": {
"certificateMode": enum,
"numRetries": integer,
"destination": {
"destinationRefId": "string"
},
"connectionRetryInterval": integer
},
"archiveOutputSettings": {
"extension": "string",
"containerSettings": {
"m2tsSettings": {
"audioStreamType": enum,
"ecmPid": "string",
"dvbTeletextPid": "string",
"aribCaptionsPidControl": enum,
"bitrate": integer,
"m2tsSettings": {
"audioStreamType": enum,
"ecmPid": "string",
"dvbTeletextPid": "string",
"aribCaptionsPidControl": enum,
"bitrate": integer,
"rateMode": enum,
"segmentationTime": number,
"audioPids": "string",
"audioFramesPerPes": integer,
"fragmentTime": number,
"ebpLookaheadMs": integer,
"ebpAudioInterval": enum,
"scte35Pid": "string",
"programNum": integer,
"pmtInterval": integer,
"pcrPeriod": integer,
"segmentationStyle": enum,
"ebif": enum,
"audioBufferSize": enum,
"dvbNitSettings": {
  "networkName": "string",
  "networkId": integer,
  "repInterval": integer
},
"absentInputAudioBehavior": enum,
"timedMetadataBehavior": enum,
"timedMetadataPid": "string",
"pmtPid": "string",
"etvSignalPid": "string",
"bufferModel": enum,
"scte35Control": enum,
"ebpPlacement": enum,
"arib": enum,
"nullPacketBitrate": number,
"dvbSdtSettings": {
  "serviceName": "string",
  "serviceProviderName": "string",
  "repInterval": integer,
  "outputSdt": enum
},
"pcrPid": "string",
"transportStreamId": integer,
"pcrControl": enum,
"videoPid": "string",
"esRateInPes": enum,
"segmentationMarkers": enum,
"dvbTdtSettings": {
  "repInterval": integer
},
"klv": enum,
"ccDescriptor": enum,
"patInterval": integer,
"etvPlatformPid": "string",
"dvbSubPids": "string",
"aribCaptionsPid": "string",
"scte27Pids": "string",
"klvDataPids": "string"
},
"nameModifier": "string"
},
"frameCaptureOutputSettings": {
  "nameModifier": "string"
},
"msSmoothOutputSettings": {
  "nameModifier": "string"
},
"mediaPackageOutputSettings": {
},
"udpOutputSettings": {
  "destination": {

"destinationRefId": "string",
"bufferMsec": integer,
"containerSettings": {
  "m2tsSettings": {
    "audioStreamType": enum,
    "ecmPid": "string",
    "dvbTeletextPid": "string",
    "aribCaptionsPidControl": enum,
    "bitrate": integer,
    "rateMode": enum,
    "segmentationTime": number,
    "audioPids": "string",
    "audioFramesPerPes": integer,
    "fragmentTime": number,
    "ebpLookaheadMs": integer,
    "ebpAudioInterval": enum,
    "scte35Pid": "string",
    "programNum": integer,
    "pmtInterval": integer,
    "pcrPeriod": integer,
    "segmentationStyle": enum,
    "ebif": enum,
    "audioBufferModel": enum,
    "dvbNitSettings": {
      "networkName": "string",
      "networkId": integer,
      "repInterval": integer
    },
    "absentInputAudioBehavior": enum,
    "timedMetadataBehavior": enum,
    "timedMetadataPid": "string",
    "pmtPid": "string",
    "etvSignalPid": "string",
    "bufferModel": enum,
    "ebpPlacement": enum,
    "ebif": enum,
    "nullPacketBitrate": number,
    "dvbSdtSettings": {
      "serviceName": "string",
      "serviceProviderName": "string",
      "repInterval": integer,
      "outputSdt": enum
    },
    "pcrPid": "string",
    "transportStreamId": integer,
    "pcrControl": enum,
    "videoPid": "string",
    "esRateInPes": enum,
    "segmentationMarkers": enum,
    "dvbTdtSettings": {
      "repInterval": integer
    },
    "klv": enum,
    "ccDescriptor": enum,
    "patInterval": integer,
    "etvPlatformPid": "string",
    "dvbSubPids": "string",
    "aribCaptionsPid": "string",
    "scte27Pids": "string",
    "klvDataPids": "string"
  }
},
"fecOutputSettings": {
  "rowLength": integer,
}
"columnDepth": integer,
"includeFec": enum
}
),
"hlsOutputSettings": {
"segmentModifier": "string",
"hlsSettings": {
"standardHlsSettings": {
"m3u8Settings": {
"pmtPid": "string",
"ecmPid": "string",
"scte35Behavior": enum,
"pcrPid": "string",
"audioPids": "string",
"audioFramesPerPes": integer,
"scte35Pid": "string",
"transportStreamId": integer,
"pcrControl": enum,
"videoPid": "string",
"pcrPeriod": integer,
"pmtInterval": enum,
"programNum": integer,
"patInterval": integer,
"timedMetadataPid": "string",
"timedMetadataBehavior": enum
},
"audioRenditionSets": "string"
},
"audioOnlyHlsSettings": {
"audioTrackType": enum,
"audioGroupId": "string",
"audioOnlyImage": {
"passwordParam": "string",
"uri": "string",
"username": "string"
}
},
"nameModifier": "string"
},
"audioDescriptionNames": [
"string"
]
},
"outputGroupSettings": {
"archiveGroupSettings": {
"destination": {
"destinationRefId": "string"
},
"rolloverInterval": integer
},
"mediaPackageGroupSettings": {
"destination": {
"destinationRefId": "string"
}
},
"rtmpGroupSettings": {
"inputLossAction": enum,
"captionData": enum,
"authenticationScheme": enum,
"cacheLength": integer,
"restartDelay": integer,
"cacheFullBehavior": enum
}.
"udpGroupSettings": {
  "inputLossAction": enum,
  "timedMetadataId3Frame": enum,
  "timedMetadataId3Period": integer
},
"msSmoothGroupSettings": {
  "fragmentLength": integer,
  "eventId": "string",
  "timestampOffset": "string",
  "segmentationMode": enum,
  "numRetries": integer,
  "eventStopBehavior": enum,
  "acquisitionPointId": "string",
  "sparseTrackType": enum,
  "timestampOffsetMode": enum,
  "destination": {
    "destinationRefId": "string"
  },
  "audioOnlyTimecodeControl": enum,
  "connectionRetryInterval": integer,
  "filecacheDuration": integer,
  "certificateMode": enum,
  "inputLossAction": enum,
  "sendDelayMs": integer,
  "eventIdMode": enum,
  "restartDelay": integer,
  "streamManifestBehavior": enum
},
"hlsGroupSettings": {
  "segmentsPerSubdirectory": integer,
  "ivInManifest": enum,
  "outputSelection": enum,
  "destination": {
    "destinationRefId": "string"
  },
  "encryptionType": enum,
  "indexNSegments": integer,
  "constantIV": "string",
  "timedMetadataId3Frame": enum,
  "baseUrlManifest": "string",
  "captionLanguageSetting": enum,
  "minSegmentLength": integer,
  "mode": enum,
  "keyProviderSettings": {
    "staticKeySettings": {
      "staticKeyValue": "string",
      "keyProviderServer": {
        "passwordParam": "string",
        "uri": "string",
        "username": "string"
      }
    }
  },
  "manifestCompression": enum,
  "ivSource": enum,
  "tsFileMode": enum,
  "manifestDurationFormat": enum,
  "keyFormatVersions": "string",
  "streamInfResolution": enum,
  "timestampDeltaMilliseconds": integer,
  "baseUrlContent": "string",
  "segmentationMode": enum,
  "captionLanguageMappings": [
  {
    "languageDescription": "string",
    "captionChannel": integer,
  }
]}
"languageCode": "string",
"clientCache": enum,
"codecSpecification": enum,
"keepSegments": integer,
"redundantManifest": enum,
"timedMetadataId3Period": integer,
"programDateTime": enum,
"directoryStructure": enum,
"keyFormat": "string",
"inputLossAction": enum,
"adMarkers": [
  enum
],
"programDateTimePeriod": integer,
"segmentLength": integer,
"hlsCdnSettings": {
  "hlsAkamaiSettings": {
    "salt": "string",
    "httpTransferMode": enum,
    "numRetries": integer,
    "restartDelay": integer,
    "connectionRetryInterval": integer,
    "fileCacheDuration": integer,
    "token": "string"
  },
  "hlsWebdavSettings": {
    "httpTransferMode": enum,
    "numRetries": integer,
    "restartDelay": integer,
    "connectionRetryInterval": integer,
    "fileCacheDuration": integer
  },
  "hlsBasicPutSettings": {
    "numRetries": integer,
    "restartDelay": integer,
    "connectionRetryInterval": integer,
    "fileCacheDuration": integer
  },
  "hlsMediaStoreSettings": {
    "mediaStoreStorageClass": enum,
    "numRetries": integer,
    "restartDelay": integer,
    "connectionRetryInterval": integer,
    "fileCacheDuration": integer
  }
},
"iFrameOnlyPlaylists": enum,
"frameCaptureGroupSettings": {
  "destination": {
    "destinationRefId": "string"
  }
},
"name": "string"
"audioDescriptions": [ 
  { 
    "audioTypeControl": enum,
    "languageCodeControl": enum,
    "remixSettings": { 
      "channelMappings": [ 
      ]
    }
  }
]
"outputChannel": integer,
"inputChannelLevels": [
    {
        "inputChannel": integer,
        "gain": integer
    }
]
"channelsOut": integer,
"channelsIn": integer
"audioType": enum,
"name": "string",
"codecSettings": {
    "aacSettings": {
        "vbrQuality": enum,
        "codingMode": enum,
        "profile": enum,
        "inputType": enum,
        "bitrate": number,
        "rawFormat": enum,
        "rateControlMode": enum,
        "sampleRate": number,
        "spec": enum
    },
    "ac3Settings": {
        "drcProfile": enum,
        "dialnorm": integer,
        "codingMode": enum,
        "metadataControl": enum,
        "bitrate": number,
        "lfeFilter": enum,
        "bitstreamMode": enum
    },
    "eac3Settings": {
        "dialnorm": integer,
        "passthroughControl": enum,
        "drcLine": enum,
        "metadataControl": enum,
        "bitrate": number,
        "ltRtSurroundMixLevel": number,
        "surroundExMode": enum,
        "lfeControl": enum,
        "codingMode": enum,
        "surroundMode": enum,
        "attenuationControl": enum,
        "lfeFilter": enum,
        "dcFilter": enum,
        "ltRtCenterMixLevel": number,
        "phaseControl": enum,
        "bitstreamMode": enum,
        "stereoDownmix": enum,
        "loRoSurroundMixLevel": number,
        "drcRF": enum,
        "loRoCenterMixLevel": number
    },
    "passThroughSettings": {
    },
    "mp2Settings": {
        "codingMode": enum,
        "bitrate": number,
        "sampleRate": number
    }
},
"languageCode": "string"
"streamName": "string",
"audioNormalizationSettings": {
  "targetLkfs": number,
  "algorithmControl": enum,
  "algorithm": enum
},
"audioSelectorName": "string"
}]
,"availConfiguration": {
  "availSettings": {
    "scte35TimeSignalApos": {
      "adAvailOffset": integer,
      "webDeliveryAllowedFlag": enum,
      "noRegionalBlackoutFlag": enum
    },
    "scte35SpliceInsert": {
      "adAvailOffset": integer,
      "webDeliveryAllowedFlag": enum,
      "noRegionalBlackoutFlag": enum
    }
  }
},
"captionDescriptions": [
  {
    "captionSelectorName": "string",
    "languageDescription": "string",
    "name": "string",
    "languageCode": "string",
    "destinationSettings": {
      "burnInDestinationSettings": {
        "xPosition": integer,
        "backgroundColor": enum,
        "yPosition": integer,
        "teletextGridControl": enum,
        "backgroundColorOpacity": integer,
        "fontOpacity": integer,
        "fontResolution": integer,
        "shadowOpacity": integer,
        "shadowYOffset": integer,
        "outlineSize": integer,
        "outlineColor": enum,
        "fontSize": "string",
        "alignment": enum,
        "shadowXOffset": integer,
        "shadowColor": enum,
        "font": {
          "passwordParam": "string",
          "uri": "string",
          "username": "string"
        }
      },
      "scte27DestinationSettings": {
      },
      "teletextDestinationSettings": {
      },
      "ttmlDestinationSettings": {
        "styleControl": enum
      },
      "smpteTtDestinationSettings": {
      },
      "webvttDestinationSettings": {
      },
      "embeddedPlusScte20DestinationSettings": {
      }
    }
  }
]
"dvbSubDestinationSettings": {
  "xPosition": integer,
  "backgroundColor": enum,
  "yPosition": integer,
  "teletextGridControl": enum,
  "backgroundOpacity": integer,
  "fontOpacity": integer,
  "fontResolution": integer,
  "shadowOpacity": integer,
  "shadowXOffset": integer,
  "outlineSize": integer,
  "outlineColor": enum,
  "fontSize": "string",
  "alignment": enum,
  "shadowYOffset": integer,
  "fontColor": enum,
  "font": {
    "passwordParam": "string",
    "uri": "string",
    "username": "string"
  }
},
"embeddedDestinationSettings": {
},
"rtmpCaptionInfoDestinationSettings": {
},
"aribDestinationSettings": {
},
"scte20PlusEmbeddedDestinationSettings": {
}
},
"globalConfiguration": {
  "inputLossBehavior": {
    "inputLossImageType": enum,
    "inputLossImageColor": "string",
    "inputLossImageSlate": {
      "passwordParam": "string",
      "uri": "string",
      "username": "string"
    },
    "repeatFrameMsec": integer,
    "blackFrameMsec": integer
  },
  "supportLowFramerateInputs": enum,
  "outputLockingMode": enum,
  "initialAudioGain": integer,
  "inputEndAction": enum,
  "outputTimingSource": enum
},
"videoDescriptions": [
  {
    "respondToAfd": enum,
    "scalingBehavior": enum,
    "name": "string",
    "width": integer,
    "sharpness": integer,
    "codecSettings": {
      "h264Settings": {
        "minIInterval": integer,
        "slices": integer,
        "parNumerator": integer,
        "gopSizeUnits": enum,
        "subgopLength": enum
      }
    }
  }
]
"maxBitrate": integer,
"bitrate": integer,
"bufFillPct": integer,
"temporalAq": enum,
"afdsignaling": enum,
"timecodeinsertion": enum,
"bufSize": integer,
"softness": integer,
"framerateControl": enum,
"qvbrQualityLevel": integer,
"fixedAfD": enum,
"level": enum,
"lookAheadRateControl": enum,
"profile": enum,
"framerate numerator": integer,
"gopClosedCadence": integer,
"entropyEncoding": enum,
"framerate denominator": integer,
"spatialAq": enum,
"adaptiveQuantization": enum,
"colorMetadata": enum,
"gopSize": number,
"numRefFrames": integer,
"gop B Reference": enum,
"parControl": enum,
"parDenominator": integer,
"syntax": enum,
"sceneChangeDetect": enum,
"scanType": enum,
"flickerAq": enum,
"gop Num B Frames": integer,
"rateControlMode": enum
},
"frameCaptureSettings": {
  "captureInterval": integer
},
"height": integer
},
"availBlanking": {
  "state": enum,
  "availBlankingImage": {
    "passwordParam": "string",
    "uri": "string",
    "username": "string"
  }
},
"blackoutSlate": {
  "network End Blackout Image": {
    "passwordParam": "string",
    "uri": "string",
    "username": "string"
  },
  "network End Blackout": enum,
  "networkId": "string",
  "state": enum,
  "blackout Slate Image": {
    "passwordParam": "string",
    "uri": "string",
    "username": "string"
  }
}
},
"InputSpecification": {
  "codec": enum,
"resolution": enum,
"maximumBitrate": enum
},
"channelClass": enum,
"tags": {
}
}

Response Bodies

Example ListChannelsResultModel

{
"channels": [
{
"inputAttachments": [
{
"inputId": "string",
"inputAttachmentName": "string",
"inputSettings": {
"sourceEndBehavior": enum,
"deblockFilter": enum,
"audioSelectors": [
{
"name": "string",
"selectorSettings": {
"audioLanguageSelection": {
"languageSelectionPolicy": enum,
"languageCode": "string"
},
"audioPidSelection": {
"pid": integer
}
}
],
"networkInputSettings": {
"hlsInputSettings": {
"retries": integer,
"bandwidth": integer,
"retryInterval": integer,
"bufferSegments": integer
},
"serverValidation": enum
},
"inputFilter": enum,
"videoSelector": {
"colorSpace": enum,
"selectorSettings": {
"videoSelectorPid": {
"pid": integer
},
"videoSelectorProgramId": {
"programId": integer
}
},
"colorSpaceUsage": enum
},
"filterStrength": integer,
"denoiseFilter": enum,
"captionSelectors": [
{
"name": "string",
"selectorSettings": {
"audioLanguageSelection": {
"languageSelectionPolicy": enum,
"languageCode": "string"
},
"audioPidSelection": {
"pid": integer
}
}
]
},
"networkInputSettings": {
"hlsInputSettings": {
"retries": integer,
"bandwidth": integer,
"retryInterval": integer,
"bufferSegments": integer
},
"serverValidation": enum
},
"inputFilter": enum,
"languageCode": "string",
"selectorSettings": {
  "embeddedSourceSettings": {
    "scte20Detection": enum,
    "source608ChannelNumber": integer,
    "convert608To708": enum,
    "source608TrackNumber": integer
  },
  "scte20SourceSettings": {
    "source608ChannelNumber": integer,
    "convert608To708": enum
  },
  "dvbSubSourceSettings": {
    "pid": integer
  },
  "teletextSourceSettings": {
    "pageNumber": "string"
  },
  "aribSourceSettings": {
  },
  "scte27SourceSettings": {
    "pid": integer
  }
},
"destinations": [
  {
    "mediaPackageSettings": [
      {
        "channelId": "string"
      }
    ],
    "settings": [
      {
        "passwordParam": "string",
        "streamName": "string",
        "url": "string",
        "username": "string"
      }
    ],
    "id": "string"
  }
],
"egressEndpoints": [
  {
    "sourceIp": "string"
  }
],
"inputSpecification": {
  "codec": enum,
  "resolution": enum,
  "maximumBitrate": enum
},
"channelClass": enum,
"tags": {
},
"logLevel": enum,
"roleArn": "string",
"name": "string",
"id": "string",
"state": enum,
"pipelinesRunningCount": integer,
Example CreateChannelResultModel

```json
{
  "channel": {
    "inputAttachments": [
      {
        "inputId": "string",
        "inputAttachmentName": "string",
        "inputSettings": {
          "sourceEndBehavior": enum,
          "deblockFilter": enum,
          "audioSelectors": [
            {
              "name": "string",
              "selectorSettings": {
                "audioLanguageSelection": {
                  "languageSelectionPolicy": enum,
                  "languageCode": "string"
                },
                "audioPidSelection": {
                  "pid": integer
                }
              }
            }
          ],
          "networkInputSettings": {
            "hlsInputSettings": {
              "retries": integer,
              "bandwidth": integer,
              "retryInterval": integer,
              "bufferSegments": integer
            },
            "serverValidation": enum
          },
          "inputFilter": enum,
          "videoSelector": {
            "colorSpace": enum,
            "selectorSettings": {
              "videoSelectorPid": {
                "pid": integer
              },
              "videoSelectorProgramId": {
                "programId": integer
              }
            },
            "colorSpaceUsage": enum
          },
          "filterStrength": integer,
          "denoiseFilter": enum,
          "captionSelectors": [
            {
              "name": "string",
              "languageCode": "string",
              "selectorSettings": {
                "embeddedSourceSettings": {
                  "scte20Detection": enum,
                  "source608ChannelNumber": integer,
                  "convert608To708": enum,
                }
              }
            }
          ]
        }
      }
    ],
    "nextToken": "string"
  }
}
```
"source608TrackNumber": integer,
"scte20SourceSettings": {
  "source608ChannelNumber": integer,
  "convert608To708": enum
},
"dvbSubSourceSettings": {
  "pid": integer
},
"teletextSourceSettings": {
  "pageNumber": "string"
},
"aribSourceSettings": {
},
"scte27SourceSettings": {
  "pid": integer
}
]
],
"destinations": [
{
  "mediaPackageSettings": [
  {
    "channelId": "string"
  }
],
  "settings": [
  {
    "passwordParam": "string",
    "streamName": "string",
    "url": "string",
    "username": "string"
  }
],
  "id": "string"
}
],
"encoderSettings": {
  "timecodeConfig": {
    "syncThreshold": integer,
    "source": enum
  },
  "outputGroups": [
  {
    "outputs": [
    {
      "videoDescriptionName": "string",
      "outputName": "string",
      "captionDescriptionNames": [
      "string"
    ],
    "outputSettings": {
      "rtmpOutputSettings": {
      "certificateMode": enum,
      "numRetries": integer,
      "destination": {
      "destinationRefId": "string"
      },
      "connectionRetryInterval": integer
      },
      "archiveOutputSettings": {
      "extension": "string",
      "archiveDir": "string"
      },
      "passwordParam": "string",
      "streamName": "string",
      "url": "string",
      "username": "string"
      }
    }
  ]
}
"containerSettings": { 
  "m2tsSettings": { 
    "audioStreamType": enum, 
    "ecmPid": "string", 
    "dvbTeletextPid": "string", 
    "aribCaptionsPidControl": enum, 
    "bitrate": integer, 
    "rateMode": enum, 
    "segmentationTime": number, 
    "audioPids": "string", 
    "audioFramesPerPes": integer, 
    "fragmentTime": number, 
    "ebpLookaheadMs": integer, 
    "ebpAudioInterval": 0, 
    "scte35Pid": "string", 
    "programNum": integer, 
    "pmtInterval": integer, 
    "pcrPeriod": integer, 
    "segmentationStyle": enum, 
    "ebp": 0, 
    "audioBufferModel": enum, 
    "dvbNlitSettings": { 
      "networkName": "string", 
      "networkId": integer, 
      "repInterval": integer 
    }, 
    "absentInputAudioBehavior": enum, 
    "timedMetadataBehavior": enum, 
    "timedMetadataPid": "string", 
    "pmtPid": "string", 
    "etvSignalPid": "string", 
    "bufferModel": enum, 
    "etv": enum, 
    "nullPacketBitrate": number, 
    "dvbSdSettings": { 
      "serviceName": "string", 
      "serviceProviderName": "string", 
      "repInterval": integer, 
      "outputSdt": enum 
    }, 
    "pcrPid": "string", 
    "transportStreamId": integer, 
    "pcrControl": enum, 
    "videoPid": "string", 
    "esRateInPes": enum, 
    "segmentationMarkers": enum, 
    "dvbTdtSettings": { 
      "repInterval": integer 
    }, 
    "klv": enum, 
    "ccDescriptor": enum, 
    "patInterval": integer, 
    "etvChannelPid": "string", 
    "dvbSubPids": "string", 
    "aribCaptionsPid": "string", 
    "scte27Pids": "string", 
    "klvDataPids": "string" 
  } 
}, 
"nameModifier": "string" 
}
"msSmoothOutputSettings": { 
  "nameModifier": "string"
},
"mediaPackageOutputSettings": {
},
"udpOutputSettings": {
  "destination": {
    "destinationRefId": "string"
  },
  "bufferMsec": integer,
  "containerSettings": {
    "m2tsSettings": {
      "audioStreamType": enum,
      "ecmPid": "string",
      "dvbTeletextPid": "string",
      "aribCaptionsPidControl": enum,
      "bitrate": integer,
      "rateMode": enum,
      "segmentationTime": number,
      "audioPids": "string",
      "audioFramesPerPes": integer,
      "fragmentTime": number,
      "ebpLookaheadMs": integer,
      "ebpAudioInterval": enum,
      "scte35Pid": "string",
      "programNum": integer,
      "pmtInterval": integer,
      "pcrPeriod": integer,
      "segmentationStyle": enum,
      "ebpPlacement": enum,
      "arib": enum,
      "nullPacketBitrate": number,
      "dvbSdtSettings": {
        "serviceName": "string",
        "serviceProviderName": "string",
        "repInterval": integer,
        "outputSdt": enum
      },
      "pcrPid": "string",
      "transportStreamId": integer,
      "pcrControl": enum,
      "videoPid": "string",
      "esRateInPes": enum,
      "segmentationMarkers": enum,
      "dvbTdtSettings": {
        "repInterval": integer
      },
      "klv": enum,
      "ccDescriptor": enum,
      "patInterval": integer,
      "etvPlatformPid": "string",
      "dvbSubPids": "string",
    }
  }
}
"aribCaptionsPid": "string",
"scte27Pids": "string",
"klvDataPids": "string"
}
},
"fecOutputSettings": {
"rowLength": integer,
"columnDepth": integer,
"includeFec": enum
}
},
"hlsOutputSettings": {
"segmentModifier": "string",
"hlsSettings": {
"standardHlsSettings": {
"m3u8Settings": {
"pmtPid": "string",
"ecmPid": "string",
"scte35Behavior": enum,
"pcrPid": "string",
"audioPids": "string",
"audioFramesPerPes": integer,
"scte35Pid": "string",
"transportStreamId": integer,
"pcrControl": enum,
"videoPid": "string",
"pcrPeriod": integer,
"pmtInterval": integer,
"programNum": integer,
"patInterval": integer,
"timedMetadataPid": "string",
"timedMetadataBehavior": enum
},
"audioRenditionSets": "string"
},
"audioOnlyHlsSettings": {
"audioTrackType": enum,
"audioGroupId": "string",
"audioOnlyImage": {
"passwordParam": "string",
"uri": "string",
"username": "string"
}
}
},
"nameModifier": "string"
}
],
"audioDescriptionNames": [
"string"
]
},
"outputGroupSettings": {
"archiveGroupSettings": {
"destination": {
"destinationRefId": "string"
},
"rolloverInterval": integer
},
"mediaPackageGroupSettings": {
"destination": {
"destinationRefId": "string"
}
},
"rtmpGroupSettings": {
}
"inputLossAction": enum,
"captionData": enum,
"authenticationScheme": enum,
"cacheLength": integer,
"restartDelay": integer,
"cacheFullBehavior": enum
},
"udpGroupSettings": {
"inputLossAction": enum,
"timedMetadataId3Frame": enum,
"timedMetadataId3Period": integer
},
"msSmoothGroupSettings": {
"fragmentLength": integer,
"eventId": "string",
"timestampOffset": "string",
"segmentationMode": enum,
"numRetries": integer,
"eventStopBehavior": enum,
"acquisitionPointId": "string",
"sparseTrackType": enum,
"timestampOffsetMode": enum,
"destination": {
"destinationRefId": "string"
},
"audioOnlyTimecodeControl": enum,
"connectionRetryInterval": integer,
"filecacheDuration": integer,
"certificateMode": enum,
"inputLossAction": enum,
"sendDelayMs": integer,
"eventIdMode": enum,
"restartDelay": integer,
"streamManifestBehavior": enum
},
"hlsGroupSettings": {
"segmentsPerSubdirectory": integer,
"ivInManifest": enum,
"outputSelection": enum,
"destination": {
"destinationRefId": "string"
},
"encryptionType": enum,
"indexNSegments": integer,
"constantIv": "string",
"baseUrlManifest": "string",
"captionLanguageSetting": enum,
"minSegmentLength": integer,
"mode": enum,
"keyProviderSettings": {
"staticKeySettings": {
"staticKeyValue": "string",
"keyProviderServer": {
"passwordParam": "string",
"uri": "string",
"username": "string"
}
}
},
"manifestCompression": enum,
"ivSource": enum,
"tsFileMode": enum,
"manifestDurationFormat": enum,
"keyFormatVersions": "string",
"streamInfResolution": enum,
"timestampDeltaMilliseconds": integer,
"baseUrlContent": "string",
"segmentationMode": enum,
"captionLanguageMappings": [
  {
    "languageDescription": "string",
    "captionChannel": integer,
    "languageCode": "string"
  }
],
"clientCache": enum,
"codecSpecification": enum,
"keepSegments": integer,
"redundantManifest": enum,
"timedMetadataId3Period": integer,
"programDateTime": enum,
"directoryStructure": enum,
"keyFormat": "string",
"inputLossAction": enum,
"adMarkers": [
  enum
],
"programDateTimePeriod": integer,
"segmentLength": integer,
"hlsCdnSettings": {
  "hlsAkamaiSettings": {
    "salt": "string",
    "httpTransferMode": enum,
    "numRetries": integer,
    "restartDelay": integer,
    "connectionRetryInterval": integer,
    "filecacheDuration": integer,
    "token": "string"
  },
  "hlsWebdavSettings": {
    "httpTransferMode": enum,
    "numRetries": integer,
    "restartDelay": integer,
    "connectionRetryInterval": integer,
    "filecacheDuration": integer
  },
  "hlsBasicPutSettings": {
    "numRetries": integer,
    "restartDelay": integer,
    "connectionRetryInterval": integer,
    "filecacheDuration": integer
  },
  "hlsMediaStoreSettings": {
    "mediaStoreStorageClass": enum,
    "numRetries": integer,
    "restartDelay": integer,
    "connectionRetryInterval": integer,
    "filecacheDuration": integer
  }
},
"iFrameOnlyPlaylists": enum,
"frameCaptureGroupSettings": {
  "destination": {
    "destinationRefId": "string"
  }
},
"name": "string"}
"audioDescriptions": [
    {
      "audioTypeControl": enum,
      "languageCodeControl": enum,
      "remixSettings": {
        "channelMappings": [
          {
            "outputChannel": integer,
            "inputChannelLevels": [
              {
                "inputChannel": integer,
                "gain": integer
              }
            ]
          }
        ],
        "channelsOut": integer,
        "channelsIn": integer
      },
      "audioType": enum,
      "name": "string",
      "codecSettings": {
        "aacSettings": {
          "vbrQuality": enum,
          "codingMode": enum,
          "profile": enum,
          "inputType": enum,
          "bitrate": number,
          "rawFormat": enum,
          "rateControlMode": enum,
          "sampleRate": number,
          "spec": enum
        },
        "ac3Settings": {
          "drcProfile": enum,
          "dialnorm": integer,
          "codingMode": enum,
          "metadataControl": enum,
          "bitrate": number,
          "lfeFilter": enum,
          "bitstreamMode": enum
        },
        "eac3Settings": {
          "dialnorm": integer,
          "passthroughControl": enum,
          "drcLine": enum,
          "metadataControl": enum,
          "bitrate": number,
          "ltRtSurroundMixLevel": number,
          "surroundExMode": enum,
          "lfeControl": enum,
          "codingMode": enum,
          "surroundMode": enum,
          "attenuationControl": enum,
          "lfeFilter": enum,
          "dcFilter": enum,
          "ltRtCenterMixLevel": number,
          "phaseControl": enum,
          "bitstreamMode": enum,
          "stereoDownmix": enum,
          "loRoSurroundMixLevel": number,
          "drcRf": enum,
          "loRoCenterMixLevel": number
        },
        "passThroughSettings": {
        }
      }
    }
]
"mp2Settings": {
  "codingMode": enum,
  "bitrate": number,
  "sampleRate": number
},
"languageCode": "string",
"streamName": "string",
"audioNormalizationSettings": {
  "targetLkfs": number,
  "algorithmControl": enum,
  "algorithm": enum
},
"audioSelectorName": "string"
},
"availConfiguration": {
  "availSettings": {
    "scte35TimeSignalApos": {
      "adAvailOffset": integer,
      "webDeliveryAllowedFlag": enum,
      "noRegionalBlackoutFlag": enum
    },
    "scte35SpliceInsert": {
      "adAvailOffset": integer,
      "webDeliveryAllowedFlag": enum,
      "noRegionalBlackoutFlag": enum
    }
  }
},
"captionDescriptions": [
  {
    "captionSelectorName": "string",
    "languageDescription": "string",
    "name": "string",
    "languageCode": "string",
    "destinationSettings": {
      "burnInDestinationSettings": {
        "xPosition": integer,
        "backgroundColor": enum,
        "yPosition": integer,
        "teletextGridControl": enum,
        "backgroundColor": integer,
        "fontOpacity": integer,
        "fontResolution": integer,
        "shadowOpacity": integer,
        "shadowYOffset": integer,
        "outlineSize": integer,
        "outlineColor": enum,
        "fontSize": "string",
        "alignment": enum,
        "shadowXOffset": integer,
        "shadowColor": integer,
        "foregroundColor": enum,
        "font": {
          "passwordParam": "string",
          "uri": "string",
          "username": "string"
        }
      },
      "scte27DestinationSettings": {
      },
      "teletextDestinationSettings": {
      },
      "ttmlDestinationSettings": {
        "styleControl": enum
      }
    }
  }
]}
"globalConfiguration": {
  "inputLossBehavior": {
    "inputLossImageType": enum,
    "inputLossImageColor": "string",
    "inputLossImageSlate": {
      "passwordParam": "string",
      "uri": "string",
      "username": "string"
    },
    "repeatFrameMsec": integer,
    "blackFrameMsec": integer
  },
  "supportLowFramerateInputs": enum,
  "outputLockingMode": enum,
  "initialAudioGain": integer,
  "inputEndAction": enum,
  "outputTimingSource": enum
},
"videoDescriptions": [
  {
    "respondToAfd": enum,
    "scalingBehavior": enum,
    "name": "string",
    "width": integer,
    "sharpness": integer,
    "xPosition": integer,
    "backgroundColor": enum,
    "yPosition": integer,
    "teletextGridControl": enum,
    "backgroundOpacity": integer,
    "fontOpacity": integer,
    "fontResolution": integer,
    "shadowOpacity": integer,
    "shadowYOffset": integer,
    "outlineSize": integer,
    "outlineColor": enum,
    "fontSize": "string",
    "alignment": enum,
    "shadowXOffset": integer,
    "shadowColor": enum,
    "fontColor": enum,
    "font": {"passwordParam": "string",
             "uri": "string",
             "username": "string"
           },
    "xPosition": integer,
    "backgroundColor": enum,
    "yPosition": integer,
    "teletextGridControl": enum,
    "backgroundOpacity": integer,
    "fontOpacity": integer,
    "fontResolution": integer,
    "shadowOpacity": integer,
    "shadowYOffset": integer,
    "outlineSize": integer,
    "outlineColor": enum,
    "fontSize": "string",
    "alignment": enum,
    "shadowXOffset": integer,
    "shadowColor": enum,
    "fontColor": enum,
    "font": {"passwordParam": "string",
             "uri": "string",
             "username": "string"
           }
  }
]
"codecSettings": {
  "h264Settings": {
    "minIInterval": integer,
    "slices": integer,
    "parNumerator": integer,
    "gopSizeUnits": enum,
    "subgopLength": enum,
    "maxBitrate": integer,
    "bitrate": integer,
    "bufFillPct": integer,
    "temporalAq": enum,
    "afdSignaling": enum,
    "timecodeInsertion": enum,
    "bufSize": integer,
    "softness": integer,
    "framerateControl": enum,
    "qvbrQualityLevel": integer,
    "fixedAfd": enum,
    "level": enum,
    "lookAheadRateControl": enum,
    "profile": enum,
    "framerateNumerator": integer,
    "gopClosedCadence": integer,
    "entropyEncoding": enum,
    "framerateDenominator": integer,
    "spatialAq": enum,
    "adaptiveQuantization": enum,
    "colorMetadata": enum,
    "gopSize": number,
    "numRefFrames": integer,
    "gopBReference": enum,
    "parControl": enum,
    "parDenominator": integer,
    "syntax": enum,
    "sceneChangeDetect": enum,
    "scanType": enum,
    "flickerAq": enum,
    "gopNumBFrames": integer,
    "rateControlMode": enum
  },
  "frameCaptureSettings": {
    "captureInterval": integer
  }
},
"height": integer
}];
"availBlanking": {
  "state": enum,
  "availBlankingImage": {
    "passwordParam": "string",
    "uri": "string",
    "username": "string"
  }
},
"blackoutSlate": {
  "networkEndBlackoutImage": {
    "passwordParam": "string",
    "uri": "string",
    "username": "string"
  },
  "networkEndBlackout": enum,
  "networkId": "string",
  "state": enum,
  "blackoutSlateImage": {
    "passwordParam": "string",
    "uri": "string",
    "username": "string"
  }
}
"uri": "string",
"username": "string"
}
},
"egressEndpoints": [
{
"sourceIp": "string"
}
],
"inputSpecification": {
"codec": enum,
"resolution": enum,
"maximumBitrate": enum
},
"channelClass": enum,
"tags": {
},
"logLevel": enum,
"roleArn": "string",
"name": "string",
"id": "string",
"state": enum,
"pipelinesRunningCount": integer,
"arn": "string"
}
}

Example InvalidRequest

{
"message": "string"
}

Example AccessDenied

{
"message": "string"
}

Example ResourceConflict

{
"message": "string"
}

Example ChannelConfigurationValidationError

{
"validationErrors": [
{
"errorMessage": "string",
"elementPath": "string"
}
],
"message": "string"
}
Example LimitExceeded

```
{
  "message": "string"
}
```

Example InternalServiceError

```
{
  "message": "string"
}
```

Example BadGatewayException

```
{
  "message": "string"
}
```

Example GatewayTimeoutException

```
{
  "message": "string"
}
```

Properties

AacCodingMode

Aac Coding Mode

- AD_RECEIVER_MIX
- CODING_MODE_1_0
- CODING_MODE_1_1
- CODING_MODE_2_0
- CODING_MODE_5_1

AacInputType

Aac Input Type

- BROADCASTER_MIXED_AD
- NORMAL

AacProfile

Aac Profile

- HEV1
- HEV2
- LC
AacRateControlMode

Aac Rate Control Mode

- CBR
- VBR

AacRawFormat

Aac Raw Format

- LATM_LOAS
- NONE

AacSettings

Aac Settings

vbrQuality

VBR Quality Level - Only used if rateControlMode is VBR.

- Type: AacVbrQuality (p. 31)
- Required: False

codingMode

Mono, Stereo, or 5.1 channel layout. Valid values depend on rate control mode and profile. The adReceiverMix 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.

- Type: AacCodingMode (p. 29)
- Required: False

profile

AAC Profile.

- Type: AacProfile (p. 29)
- Required: False

inputType

Set to "broadcasterMixedAd" when input contains pre-mixed main audio + AD (narration) as a stereo pair. The Audio Type field (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. The values in audioTypeControl and audioType (in AudioDescription) are ignored when set to broadcasterMixedAd. Leave set to "normal" when input does not contain pre-mixed audio + AD.

- Type: AacInputType (p. 29)
- Required: False
**bitrate**
Average bitrate in bits/second. Valid values depend on rate control mode and profile.

Type: number  
Required: False

**rawFormat**
Sets LATM / LOAS AAC output for raw containers.

Type: AacRawFormat (p. 30)  
Required: False

**rateControlMode**
Rate Control Mode.

Type: AacRateControlMode (p. 30)  
Required: False

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

Type: number  
Required: False

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

Type: AacSpec (p. 31)  
Required: False

**AacSpec**
Aac Spec

  MPEG2  
  MPEG4

**AacVbrQuality**
Aac Vbr Quality

  HIGH  
  LOW  
  MEDIUM_HIGH  
  MEDIUM_LOW

**Ac3BitstreamMode**
Ac3 Bitstream Mode
COMMENTARY
COMPLETE_MAIN
DIALOGUE
EMERGENCY
HEARING_IMPAIRED
MUSIC_AND_EFFECTS
VISUALLY_IMPAIRED
VOICE_OVER

**Ac3CodingMode**

Ac3 Coding Mode

- CODING_MODE_1_0
- CODING_MODE_1_1
- CODING_MODE_2_0
- CODING_MODE_3_2_LFE

**Ac3DrcProfile**

Ac3 Drc Profile

- FILM_STANDARD
- NONE

**Ac3LfeFilter**

Ac3 Lfe Filter

- DISABLED
- ENABLED

**Ac3MetadataControl**

Ac3 Metadata Control

- FOLLOW_INPUT
- USE_CONFIGURED

**Ac3Settings**

Ac3 Settings

**drcProfile**

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

*Type: Ac3DrcProfile (p. 32)  
*Required: False*
**dialnorm**

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

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

**codingMode**

Dolby Digital coding mode. Determines number of channels.

- **Type**: Ac3CodingMode (p. 32)
- **Required**: False

**metadataControl**

When set to "followInput", 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.

- **Type**: Ac3MetadataControl (p. 32)
- **Required**: False

**bitrate**

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

- **Type**: number
- **Required**: False

**lfeFilter**

When set to enabled, applies a 120Hz lowpass filter to the LFE channel prior to encoding. Only valid in codingMode32Lfe mode.

- **Type**: Ac3LfeFilter (p. 32)
- **Required**: False

**bitstreamMode**

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

- **Type**: Ac3BitstreamMode (p. 31)
- **Required**: False

**AccessDenied**

**message**

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

### AfdSignaling

Afd Signaling

- AUTO
- FIXED
- NONE

### ArchiveContainerSettings

Archive Container Settings

#### m2tsSettings

- **Type**: M2tsSettings (p. 107)
- **Required**: False

### ArchiveGroupSettings

Archive Group Settings

#### destination

A directory and base filename where archive files should be written.

- **Type**: OutputLocationRef (p. 125)
- **Required**: True

#### rolloverInterval

Number of seconds to write to archive file before closing and starting a new one.

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

### ArchiveOutputSettings

Archive Output Settings

#### extension

Output file extension. If excluded, this will be auto-selected from the container type.

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

#### containerSettings

Settings specific to the container type of the file.
Properties

**Type**: ArchiveContainerSettings (p. 34)
**Required**: True

**nameModifier**

String concatenated to the end of the destination filename. Required for multiple outputs of the same type.

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

**AribDestinationSettings**

Arib Destination Settings

**AribSourceSettings**

Arib Source Settings

**AudioChannelMapping**

Audio Channel Mapping

**outputChannel**

The index of the output channel being produced.

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

**inputChannelLevels**

Indices and gain values for each input channel that should be remixed into this output channel.

**Type**: Array of type InputChannelLevel (p. 98)
**Required**: True

**AudioCodecSettings**

Audio Codec Settings

**aacSettings**

**Type**: AacSettings (p. 30)
**Required**: False

**ac3Settings**

**Type**: Ac3Settings (p. 32)
Required: False

eac3Settings

Type: Eac3Settings (p. 65)
Required: False

passThroughSettings

Type: PassThroughSettings (p. 126)
Required: False

mp2Settings

Type: Mp2Settings (p. 118)
Required: False

AudioDescription

Audio Description

audioTypeControl

Determines how audio type is determined. followInput: If the input contains an ISO 639 audioType, then that value is passed through to the output. If the input contains no ISO 639 audioType, the value in Audio Type is included in the output. useConfigured: The value in Audio Type is included in the output. Note that this field and audioType are both ignored if inputType is broadcasterMixedAd.

Type: AudioDescriptionAudioTypeControl (p. 37)
Required: False

languageCodeControl

Choosing followInput will cause the ISO 639 language code of the output to follow the ISO 639 language code of the input. The languageCode will be used when useConfigured is set, or when followInput is selected but there is no ISO 639 language code specified by the input.

Type: AudioDescriptionLanguageCodeControl (p. 38)
Required: False

remixSettings

Settings that control how input audio channels are remixed into the output audio channels.

Type: RemixSettings (p. 126)
Required: False

audioType

Applies only if audioTypeControl is useConfigured. The values for audioType are defined in ISO-IEC 13818-1.

Type: AudioType (p. 41)
name

The name of this AudioDescription. Outputs will use this name to uniquely identify this AudioDescription. Description names should be unique within this Live Event.

Type: string
Required: True

codecSettings

Audio codec settings.

Type: AudioCodecSettings (p. 35)
Required: False

languageCode

Indicates the language of the audio output track. Only used if languageControlMode is useConfigured, or there is no ISO 639 language code specified in the input.

Type: string
Required: False
MinLength: 3
MaxLength: 3

streamName

Used for MS Smooth and Apple HLS outputs. Indicates the name displayed by the player (eg. English, or Director Commentary).

Type: string
Required: False

audioNormalizationSettings

Advanced audio normalization settings.

Type: AudioNormalizationSettings (p. 39)
Required: False

audioSelectorName

The name of the AudioSelector used as the source for this AudioDescription.

Type: string
Required: True

AudioDescriptionAudioTypeControl

Audio Description Audio Type Control
FOLLOW_INPUT
USE_CONFIGURED

AudioDescriptionLanguageCodeControl

Audio Description Language Code Control

FOLLOW_INPUT
USE_CONFIGURED

AudioLanguageSelection

Audio Language Selection

languageSelectionPolicy

When set to "strict", the transport stream demux strictly identifies audio streams by their language descriptor. If a PMT update occurs such that an audio stream matching the initially selected language is no longer present then mute will be encoded until the language returns. If "loose", then on a PMT update the demux will choose another audio stream in the program with the same stream type if it can't find one with the same language.

Type: AudioLanguageSelectionPolicy (p. 38)
Required: False

languageCode

Selects a specific three-letter language code from within an audio source.

Type: string
Required: True

AudioLanguageSelectionPolicy

Audio Language Selection Policy

LOOSE
STRICT

AudioNormalizationAlgorithm

Audio Normalization Algorithm

ITU_1770_1
ITU_1770_2

AudioNormalizationAlgorithmControl

Audio Normalization Algorithm Control

CORRECT_AUDIO
AudioNormalizationSettings

Audio Normalization Settings

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
Minimum: -59.0
Maximum: 0.0

algorithmControl

When set to correctAudio the output audio is corrected using the chosen algorithm. If set to measureOnly, the audio will be measured but not adjusted.

Type: AudioNormalizationAlgorithmControl (p. 38)
Required: False

algorithm

Audio normalization algorithm to use. itu17701 conforms to the CALM Act specification, itu17702 conforms to the EBU R-128 specification.

Type: AudioNormalizationAlgorithm (p. 38)
Required: False

AudioOnlyHlsSettings

Audio Only Hls Settings

audioTrackType

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=NO, AUTOSELECT=NO Alternate Audio, Auto Select, Not Default 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

Type: AudioOnlyHlsTrackType (p. 40)
Required: False

audioGroupId

Specifies the group to which the audio Rendition belongs.

Type: string
**Properties**

**Required**: False

**audioOnlyImage**

For use with an audio only Stream. Must be a .jpg or .png file. If given, this image will be used as the cover-art for the audio only output. Ideally, it should be formatted for an iPhone screen for two reasons. The iPhone does not resize the image, it crops a centered image on the top/bottom and left/right. Additionally, this image file gets saved bit-for-bit into every 10-second segment file, so will increase bandwidth by \( \text{image file size} \times \text{segment count} \times \text{user count.} \).

*Type: InputLocation (p. 99)*  
*Required: False*

**AudioOnlyHlsTrackType**

Audio Only Hls Track Type

- ALTERNATE_AUDIO_AUTO_SELECT
- ALTERNATE_AUDIO_AUTO_SELECT_DEFAULT
- ALTERNATE_AUDIO_NOT_AUTO_SELECT
- AUDIO_ONLY_VARIANT_STREAM

**AudioPidSelection**

Audio Pid Selection

**pid**

Selects a specific PID from within a source.

*Type: integer*  
*Required: True*  
*Minimum: 0*  
*Maximum: 8191*

**AudioSelector**

Audio Selector

**name**

The name of this AudioSelector. AudioDescriptions will use this name to uniquely identify this Selector. Selector names should be unique per input.

*Type: string*  
*Required: True*  
*MinLength: 1*

**selectorSettings**

The audio selector settings.

*Type: AudioSelectorSettings (p. 41)*  
*Required: False*
AudioSelectorSettings

Audio Selector Settings

audioLanguageSelection

Type: AudioLanguageSelection (p. 38)
Required: False

audioPidSelection

Type: AudioPidSelection (p. 40)
Required: False

AudioType

Audio Type

CLEAN_EFFECTS
HEARING_IMPAIRED
UNDEFINED
VISUAL_IMPAIREDCOMMENTARY

AuthenticationScheme

Authentication Scheme

AKAMAI
COMMON

AvailBlanking

Avail Blanking

state

When set to enabled, causes video, audio and captions to be blanked when insertion metadata is added.

Type: AvailBlankingState (p. 41)
Required: False

availBlankingImage

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

Type: InputLocation (p. 99)
Required: False

AvailBlankingState

Avail Blanking State

DISABLED
ENABLED

**AvailConfiguration**
Avail Configuration

**availSettings**
Ad avail settings.

  **Type:** AvailSettings (p. 42)
  **Required:** False

**AvailSettings**
Avail Settings

**scte35TimeSignalApos**

  **Type:** Scte35TimeSignalApos (p. 131)
  **Required:** False

**scte35SpliceInsert**

  **Type:** Scte35SpliceInsert (p. 130)
  **Required:** False

**BadGatewayException**

**message**

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

**BlackoutSlate**
Blackout Slate

**networkEndBlackoutImage**
Path to local file to use as Network End Blackout image. Image will be scaled to fill the entire output raster.

  **Type:** InputLocation (p. 99)
  **Required:** False

**networkEndBlackout**
Setting to enabled causes the encoder to blackout the video, audio, and captions, and raise the "Network Blackout Image" slate when an SCTE104/35 Network End Segmentation Descriptor is encountered. The blackout will be lifted when the Network Start Segmentation Descriptor is encountered. The Network End and Network Start descriptors must contain a network ID that matches the value entered in "Network ID".
**Properties**

**Type:** BlackoutSlateNetworkEndBlackout (p. 43)  
**Required:** False

**networkId**

Provides Network ID that matches EIDR ID format (e.g., "10.XXXX/XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-C").

**Type:** string  
**Required:** False  
**MinLength:** 34  
**MaxLength:** 34

**state**

When set to enabled, causes video, audio and captions to be blanked when indicated by program metadata.

**Type:** BlackoutSlateState (p. 43)  
**Required:** False

**blackoutSlateImage**

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

**Type:** InputLocation (p. 99)  
**Required:** False

**BlackoutSlateNetworkEndBlackout**

Blackout Slate Network End Blackout

- DISABLED
- ENABLED

**BlackoutSlateState**

Blackout Slate State

- DISABLED
- ENABLED

**BurnInAlignment**

Burn In Alignment

- CENTERED
- LEFT
- SMART

**BurnInBackgroundColor**

Burn In Background Color
BurnInDestinationSettings

Burn In Destination Settings

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 xPosition is provided, the horizontal caption position will be determined by the alignment parameter. All burn-in and DVB-Sub font settings must match.

Type: integer
Required: False
Minimum: 0

backgroundColor

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

Type: BurnInBackgroundColor (p. 43)
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 yPosition is provided, the caption will be positioned towards the bottom of the output. All burn-in and DVB-Sub font settings must match.

Type: integer
Required: False
Minimum: 0

teletextGridControl

Controls whether a fixed grid size will be used to generate the output subtitles bitmap. Only applicable for Teletext inputs and DVB-Sub/Burn-in outputs.

Type: BurnInTeletextGridControl (p. 47)
Required: False

backgroundOpacity

Specifies the opacity of the background rectangle. 255 is opaque; 0 is transparent. Leaving this parameter out 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: False
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 out 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

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: 0
Maximum: 10

outlineColor

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.
Properties

**Type:** BurnInOutlineColor (p. 47)  
**Required:** False

**fontSize**

When set to 'auto' fontSize will scale depending on the size of the output. Giving a positive integer will specify the exact font size in points. All burn-in and DVB-Sub font settings must match.

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

**alignment**

If no explicit xPosition or yPosition 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. Selecting "smart" justification will left-justify live subtitles and center-justify pre-recorded subtitles. All burn-in and DVB-Sub font settings must match.

**Type:** BurnInAlignment (p. 43)  
**Required:** False

**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

**shadowColor**

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

**Type:** BurnInShadowColor (p. 47)  
**Required:** False

**fontColor**

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.

**Type:** BurnInFontColor (p. 47)  
**Required:** False

**font**

External font file used for caption burn-in. File extension must be 'ttf' or 'tte'. Although the user can select output fonts for many different types of input captions, embedded, STL and teletext sources use a strict grid system. Using external fonts with these caption sources could cause unexpected display of proportional fonts. All burn-in and DVB-Sub font settings must match.
**BurnInFontColor**

Burn In Font Color

BLACK
BLUE
GREEN
RED
WHITE
YELLOW

**BurnInOutlineColor**

Burn In Outline Color

BLACK
BLUE
GREEN
RED
WHITE
YELLOW

**BurnInShadowColor**

Burn In Shadow Color

BLACK
NONE
WHITE

**BurnInTeletextGridControl**

Burn In Teletext Grid Control

FIXED
SCALED

**CaptionDescription**

Output groups for this Live Event. Output groups contain information about where streams should be distributed.

**captionSelectorName**

Specifies which input caption selector to use as a caption source when generating output captions. This field should match a captionSelector name.

**Type:** string
Properties

Required: True

languageDescription
Human readable information to indicate captions available for players (eg. English, or Spanish).

Type: string
Required: False

name
Name of the caption description. Used to associate a caption description with an output. Names must be unique within an event.

Type: string
Required: True

languageCode

Type: string
Required: False

destinationSettings
Additional settings for captions destination that depend on the destination type.

Type: CaptionDestinationSettings (p. 48)
Required: False

CaptionDestinationSettings
Caption Destination Settings

burnInDestinationSettings

Type: BurnInDestinationSettings (p. 44)
Required: False

scte27DestinationSettings

Type: Scte27DestinationSettings (p. 129)
Required: False

teletextDestinationSettings

Type: TeletextDestinationSettings (p. 133)
Required: False

ttmlDestinationSettings

Type: TtmlDestinationSettings (p. 134)
Required: False

**smpteTtDestinationSettings**

*Type:* SmpteTtDestinationSettings (p. 133)
*Required:* False

**webvttDestinationSettings**

*Type:* WebvttDestinationSettings (p. 140)
*Required:* False

**embeddedPlusScte20DestinationSettings**

*Type:* EmbeddedPlusScte20DestinationSettings (p. 68)
*Required:* False

**dvbSubDestinationSettings**

*Type:* DvbSubDestinationSettings (p. 59)
*Required:* False

**embeddedDestinationSettings**

*Type:* EmbeddedDestinationSettings (p. 68)
*Required:* False

**rtmpCaptionInfoDestinationSettings**

*Type:* RtmpCaptionInfoDestinationSettings (p. 127)
*Required:* False

**aribDestinationSettings**

*Type:* AribDestinationSettings (p. 35)
*Required:* False

**scte20PlusEmbeddedDestinationSettings**

*Type:* Scte20PlusEmbeddedDestinationSettings (p. 129)
*Required:* False

**CaptionLanguageMapping**

Maps a caption channel to an ISO 693-2 language code (http://www.loc.gov/standards/iso639-2), with an optional description.

**languageDescription**

Textual description of language

*Type:* string
Properties

Required: True
MinLength: 1

**captionChannel**

The closed caption channel being described by this CaptionLanguageMapping. Each channel mapping must have a unique channel number (maximum of 4)

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

**languageCode**

Three character ISO 639-2 language code (see http://www.loc.gov/standards/iso639-2)

Type: string
Required: True
MinLength: 3
MaxLength: 3

**CaptionSelector**

Output groups for this Live Event. Output groups contain information about where streams should be distributed.

**name**

Name identifier for a caption selector. This name is used to associate this caption selector with one or more caption descriptions. Names must be unique within an event.

Type: string
Required: True
MinLength: 1

**languageCode**

When specified this field indicates the three letter language code of the caption track to extract from the source.

Type: string
Required: False

**selectorSettings**

Caption selector settings.

Type: CaptionSelectorSettings (p. 50)
Required: False

**CaptionSelectorSettings**

Caption Selector Settings
**embeddedSourceSettings**

Type: EmbeddedSourceSettings (p. 69)  
Required: False

**scte20SourceSettings**

Type: Scte20SourceSettings (p. 129)  
Required: False

**dvbSubSourceSettings**

Type: DvbSubSourceSettings (p. 63)  
Required: False

**teletextSourceSettings**

Type: TeletextSourceSettings (p. 133)  
Required: False

**aribSourceSettings**

Type: AribSourceSettings (p. 35)  
Required: False

**scte27SourceSettings**

Type: Scte27SourceSettings (p. 129)  
Required: False

**Channel**

**inputAttachments**

List of input attachments for channel.  
Type: Array of type InputAttachment (p. 97)  
Required: False

**destinations**

A list of destinations of the channel. For UDP outputs, there is one destination per output. For other types (HLS, for example), there is one destination per packager.  
Type: Array of type OutputDestination (p. 123)  
Required: False

**encoderSettings**

Type: EncoderSettings (p. 69)  
Required: False
egressEndpoints
The endpoints where outgoing connections initiate from
Type: Array of type ChannelEgressEndpoint (p. 53)
Required: False

inputSpecification
Type: InputSpecification (p. 103)
Required: False

channelClass
The class for this channel. STANDARD for a channel with two pipelines or SINGLEPIPELINE for a channel with one pipeline.
Type: ChannelClass (p. 53)
Required: False

tags
A collection of key-value pairs.
Type: Tags (p. 133)
Required: False

logLevel
The log level being written to CloudWatch Logs.
Type: LogLevel (p. 104)
Required: False

roleArn
The Amazon Resource Name (ARN) of the role assumed when running the Channel.
Type: string
Required: False

name
The name of the channel. (user-mutable)
Type: string
Required: False

id
The unique id of the channel.
Type: string
Required: False
state

Type: ChannelState (p. 53)
Required: False

pipelinesRunningCount

The number of currently healthy pipelines.

Type: integer
Required: False

arn

The unique arn of the channel.

Type: string
Required: False

ChannelClass

A standard channel has two encoding pipelines and a single pipeline channel only has one.

STANDARD
SINGLE_PIPELINE

ChannelConfigurationValidationError

validationErrors

A collection of validation error responses.

Type: Array of type ValidationError (p. 136)
Required: False

message

Type: string
Required: False

ChannelEgressEndpoint

sourceIp

Public IP of where a channel's output comes from

Type: string
Required: False

ChannelState

CREATING
CREATE_FAILED
IDLE
STARTING
RUNNING
RECOVERING
STOPPING
DELETING
DELETED
UPDATING
UPDATE_FAILED

ChannelSummary

inputAttachments
List of input attachments for channel.

  Type: Array of type InputAttachment (p. 97)
  Required: False

destinations
A list of destinations of the channel. For UDP outputs, there is one destination per output. For other types (HLS, for example), there is one destination per packager.

  Type: Array of type OutputDestination (p. 123)
  Required: False

egressEndpoints
The endpoints where outgoing connections initiate from

  Type: Array of type ChannelEgressEndpoint (p. 53)
  Required: False

inputSpecification

  Type: InputSpecification (p. 103)
  Required: False

channelClass
The class for this channel. STANDARD for a channel with two pipelines or SINGLE_PIPELINE for a channel with one pipeline.

  Type: ChannelClass (p. 53)
  Required: False

tags
A collection of key-value pairs.

  Type: Tags (p. 133)
**Properties**

**logLevel**

The log level being written to CloudWatch Logs.

*Type: LogLevel (p. 104)*

*Required: False*

**roleArn**

The Amazon Resource Name (ARN) of the role assumed when running the Channel.

*Type: string*

*Required: False*

**name**

The name of the channel. (user-mutable)

*Type: string*

*Required: False*

**id**

The unique id of the channel.

*Type: string*

*Required: False*

**state**

*Type: ChannelState (p. 53)*

*Required: False*

**pipelinesRunningCount**

The number of currently healthy pipelines.

*Type: integer*

*Required: False*

**arn**

The unique arn of the channel.

*Type: string*

*Required: False*

**CreateChannel**

**inputAttachments**

List of input attachments for channel.
Properties

**Type**: Array of type *InputAttachment (p. 97)*

**Required**: False

**logLevel**

The log level to write to CloudWatch Logs.

**Type**: *LogLevel (p. 104)*

**Required**: False

**reserved**

Deprecated field that's only usable by whitelisted customers.

**Type**: string

**Required**: False

**requestId**

Unique request ID to be specified. This is needed to prevent retries from creating multiple resources.

**Type**: string

**Required**: False

**roleArn**

An optional Amazon Resource Name (ARN) of the role to assume when running the Channel.

**Type**: string

**Required**: False

**destinations**

**Type**: Array of type *OutputDestination (p. 123)*

**Required**: False

**name**

Name of channel.

**Type**: string

**Required**: False

**encoderSettings**

**Type**: *EncoderSettings (p. 69)*

**Required**: False

**inputSpecification**

Specification of input for this channel (max. bitrate, resolution, codec, etc.)

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

channelClass

The class for this channel. STANDARD for a channel with two pipelines or SINGLE_PIPELINE for a channel with one pipeline.

Type: ChannelClass (p. 53)
Required: False

tags

A collection of key-value pairs.

Type: Tags (p. 133)
Required: False

CreateChannelResultModel

channel

Type: Channel (p. 51)
Required: False

DvbNitSettings

DVB Network Information Table (NIT)

networkName

The network name text placed in the networkNameDescriptor inside the Network Information Table. Maximum length is 256 characters.

Type: string
Required: True
MinLength: 1
MaxLength: 256

networkId

The numeric value placed in the Network Information Table (NIT).

Type: integer
Required: True
Minimum: 0
Maximum: 65536

repInterval

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

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

**DvbSdtOutputSdt**

Dvb Sdt Output Sdt

- SDT_FOLLOW
- SDT_FOLLOW_IF_PRESENT
- SDT_MANUAL
- SDT_NONE

**DvbSdtSettings**

DVB Service Description Table (SDT)

**serviceName**

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

  - **Type:** string
  - **Required:** False
  - **MinLength:** 1
  - **MaxLength:** 256

**serviceProviderName**

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

  - **Type:** string
  - **Required:** False
  - **MinLength:** 1
  - **MaxLength:** 256

**repInterval**

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

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

**outputSdt**

Selects method of inserting SDT information into output stream. The sdftFollow setting copies SDT information from input stream to output stream. The sdftFollowIfPresent setting 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. The sdftManual setting means user will enter the SDT information. The sdftNone setting means output stream will not contain SDT information.

  - **Type:** DvbSdtOutputSdt (p. 58)
  - **Required:** False
**DvbSubDestinationAlignment**

Dvb Sub Destination Alignment

- CENTERED
- LEFT
- SMART

**DvbSubDestinationBackgroundColor**

Dvb Sub Destination Background Color

- BLACK
- NONE
- WHITE

**DvbSubDestinationFontColor**

Dvb Sub Destination Font Color

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

**DvbSubDestinationOutlineColor**

Dvb Sub Destination Outline Color

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

**DvbSubDestinationSettings**

Dvb Sub Destination Settings

**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 xPosition 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
backgroundColor

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

Type: DvbSubDestinationBackgroundColor (p. 59)
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 yPosition 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

teletextGridControl

Controls whether a fixed grid size will be used to generate the output subtitles bitmap. Only applicable for Teletext inputs and DVB-Sub/Burn-in outputs.

Type: DvbSubDestinationTeletextGridControl (p. 62)
Required: False

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: False
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

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**: 0
- **Maximum**: 10

outlineColor

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.

- **Type**: DvbSubDestinationOutlineColor (p. 59)
- **Required**: False

fontSize

When set to auto fontSize will scale depending on the size of the output. Giving a positive integer will specify the exact font size in points. All burn-in and DVB-Sub font settings must match.

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

alignment

If no explicit xPosition or yPosition 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. Selecting "smart" justification will left-justify live subtitles and center-justify pre-recorded subtitles. This option is not valid for source captions that are STL or 608/embedded. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

**Type:** DvbSubDestinationAlignment (p. 59)
**Required:** False

### 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

### shadowColor

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

**Type:** DvbSubDestinationShadowColor (p. 62)
**Required:** False

### fontColor

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.

**Type:** DvbSubDestinationFontColor (p. 59)
**Required:** False

### font

External font file used for caption burn-in. File extension must be 'ttf' or 'tte'. Although the user can select output fonts for many different types of input captions, embedded, STL and teletext sources use a strict grid system. Using external fonts with these caption sources could cause unexpected display of proportional fonts. All burn-in and DVB-Sub font settings must match.

**Type:** InputLocation (p. 99)
**Required:** False

### DvbSubDestinationShadowColor

Dvb Sub Destination Shadow Color

- BLACK
- NONE
- WHITE

### DvbSubDestinationTeletextGridControl

Dvb Sub Destination Teletext Grid Control
DvbSubSourceSettings

Dvb Sub Source Settings

**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

DvbTdtSettings

DVB Time and Date Table (SDT)

**replInterval**

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

- **Type**: integer
- **Required**: False
- **Minimum**: 1000
- **Maximum**: 30000

Eac3AttenuationControl

Eac3 Attenuation Control

- ATTENUATE_3_DB
- NONE

Eac3BitstreamMode

Eac3 Bitstream Mode

- COMMENTARY
- COMPLETE_MAIN
- EMERGENCY
- HEARING_IMPAIRED
- VISUALLY_IMPAIRED

Eac3CodingMode

Eac3 Coding Mode

- CODING_MODE_1_0
- CODING_MODE_2_0
CODING_MODE_3_2

**Eac3DcFilter**

Eac3 Dc Filter

- DISABLED
- ENABLED

**Eac3DrcLine**

Eac3 Drc Line

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

**Eac3DrcRf**

Eac3 Drc Rf

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

**Eac3LfeControl**

Eac3 Lfe Control

- LFE
- NO_LFE

**Eac3LfeFilter**

Eac3 Lfe Filter

- DISABLED
- ENABLED

**Eac3MetadataControl**

Eac3 Metadata Control

- FOLLOW_INPUT
USE_CONFIGURED

**Eac3PassthroughControl**

Eac3 Passthrough Control

- NO_PASSTHROUGH
- WHEN_POSSIBLE

**Eac3PhaseControl**

Eac3 Phase Control

- NO_SHIFT
- SHIFT_90_DEGREES

**Eac3Settings**

Eac3 Settings

**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**

When set to whenPossible, 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.

- **Type**: Eac3PassthroughControl (p. 65)
- **Required**: False

**drcLine**

Sets the Dolby dynamic range compression profile.

- **Type**: Eac3DrcLine (p. 64)
- **Required**: False

**metadataControl**

When set to followInput, 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.

- **Type**: Eac3MetadataControl (p. 64)
- **Required**: False
Properties

**bitrate**

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

- **Type:** number
- **Required:** False

**LtRtSurroundMixLevel**

Left total/Right total surround mix level. Only used for 3/2 coding mode.

- **Type:** number
- **Required:** False

**surroundExMode**

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

- **Type:** `Eac3SurroundExMode` (p. 68)  
- **Required:** False

**lfeControl**

When encoding 3/2 audio, setting to lfe enables the LFE channel

- **Type:** `Eac3LfeControl` (p. 64)  
- **Required:** False

**codingMode**

Dolby Digital Plus coding mode. Determines number of channels.

- **Type:** `Eac3CodingMode` (p. 63)  
- **Required:** False

**surroundMode**

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

- **Type:** `Eac3SurroundMode` (p. 68)  
- **Required:** False

**attenuationControl**

When set to attenuate3Db, applies a 3 dB attenuation to the surround channels. Only used for 3/2 coding mode.

- **Type:** `Eac3AttenuationControl` (p. 63)  
- **Required:** False

**lfeFilter**

When set to enabled, applies a 120Hz lowpass filter to the LFE channel prior to encoding. Only valid with codingMode32 coding mode.
Properties

Type: Eac3LfeFilter (p. 64)
Required: False

dcFilter
When set to enabled, activates a DC highpass filter for all input channels.

Type: Eac3DcFilter (p. 64)
Required: False

ltRtCenterMixLevel
Left total/Right total center mix level. Only used for 3/2 coding mode.

Type: number
Required: False

phaseControl
When set to shift90Degrees, applies a 90-degree phase shift to the surround channels. Only used for 3/2 coding mode.

Type: Eac3PhaseControl (p. 65)
Required: False

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

Type: Eac3BitstreamMode (p. 63)
Required: False

stereoDownmix
Stereo downmix preference. Only used for 3/2 coding mode.

Type: Eac3StereoDownmix (p. 68)
Required: False

loRoSurroundMixLevel
Left only/Right only surround mix level. Only used for 3/2 coding mode.

Type: number
Required: False

drcRf
Sets the profile for heavy Dolby dynamic range compression, ensures that the instantaneous signal peaks do not exceed specified levels.

Type: Eac3DrcRf (p. 64)
Required: False
**IoRoCenterMixLevel**

Left only/Right only center mix level. Only used for 3/2 coding mode.

- **Type:** number
- **Required:** False

**Eac3StereoDownmix**

Eac3 Stereo Downmix

- DPL2
- LO_RO
- LT_RT
- NOT_INDICATED

**Eac3SurroundExMode**

Eac3 Surround Ex Mode

- DISABLED
- ENABLED
- NOT_INDICATED

**Eac3SurroundMode**

Eac3 Surround Mode

- DISABLED
- ENABLED
- NOT_INDICATED

**EmbeddedConvert608To708**

Embedded Convert608 To708

- DISABLED
- UPCONVERT

**EmbeddedDestinationSettings**

Embedded Destination Settings

**EmbeddedPlusScte20DestinationSettings**

Embedded Plus Scte20 Destination Settings

**EmbeddedScte20Detection**

Embedded Scte20 Detection

- AUTO
EmbeddedSourceSettings

Embedded Source Settings

scte20Detection
Set to "auto" to handle streams with intermittent and/or non-aligned SCTE-20 and Embedded captions.

Type: EmbeddedScte20Detection (p. 68)
Required: False

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

custom608To708
If 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.

Type: EmbeddedConvert608To708 (p. 68)
Required: False

source608TrackNumber
This field is unused and deprecated.

Type: integer
Required: False
Minimum: 1
Maximum: 5

EncoderSettings

Encoder Settings

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

Type: TimecodeConfig (p. 134)
Required: True

outputGroups

Type: Array of type OutputGroup (p. 124)
Required: True

audioDescriptions
Type: Array of type AudioDescription (p. 36)
Required: True

availConfiguration
Event-wide configuration settings for ad avail insertion.
Type: AvailConfiguration (p. 42)
Required: False

captionDescriptions
Settings for caption descriptions
Type: Array of type CaptionDescription (p. 47)
Required: False

globalConfiguration
Configuration settings that apply to the event as a whole.
Type: GlobalConfiguration (p. 72)
Required: False

videoDescriptions
Type: Array of type VideoDescription (p. 137)
Required: True

availBlanking
Settings for ad avail blanking.
Type: AvailBlanking (p. 41)
Required: False

blackoutSlate
Settings for blackout slate.
Type: BlackoutSlate (p. 42)
Required: False

FecOutputIncludeFec
Fec Output Include Fec
COLUMN
COLUMN_AND_ROW
### FecOutputSettings

**Fec Output Settings**

#### rowLength

Parameter L from SMPTE 2022-1. The width of the FEC protection matrix. Must be between 1 and 20, inclusive. If only Column FEC is used, then larger values increase robustness. If Row FEC is used, then this is the number of transport stream packets per row error correction packet, and the value must be between 4 and 20, inclusive, if includeFec is columnAndRow. If includeFec is column, this value must be 1 to 20, inclusive.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 20

#### columnDepth

Parameter D from SMPTE 2022-1. The height of the FEC protection matrix. The number of transport stream packets per column error correction packet. Must be between 4 and 20, inclusive.

- **Type:** integer
- **Required:** False
- **Minimum:** 4
- **Maximum:** 20

#### includeFec

Enables column only or column and row based FEC

- **Type:** FecOutputIncludeFec (p. 70)
- **Required:** False

### FixedAfd

**Fixed Afd**

- AFD_0000
- AFD_0010
- AFD_0011
- AFD_0100
- AFD_1000
- AFD_1001
- AFD_1010
- AFD_1011
- AFD_1101
- AFD_1110
- AFD_1111

### FrameCaptureGroupSettings

**Frame Capture Group Settings**
**destination**

The destination for the frame capture files. Either the URI for an Amazon S3 bucket and object, plus a file name prefix (for example, s3ssl://sportsDelivery/highlights/20180820/curling_) or the URI for a MediaStore container, plus a file name prefix (for example, mediastoressl://sportsDelivery/20180820/curling_). The final file names consist of the prefix from the destination field (for example, "curling_") + name modifier + the counter (5 digits, starting from 00001) + extension (which is always .jpg). For example, curlingLow.00001.jpg

- **Type**: OutputLocationRef (p. 125)
- **Required**: True

**FrameCaptureOutputSettings**

Frame Capture Output Settings

**nameModifier**

Required if the output group contains more than one output. This modifier forms part of the output file name.

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

**FrameCaptureSettings**

Frame Capture Settings

**captureInterval**

The frequency, in seconds, for capturing frames for inclusion in the output. For example, "10" means capture a frame every 10 seconds.

- **Type**: integer
- **Required**: True
- **Minimum**: 1
- **Maximum**: 3600

**GatewayTimeoutException**

**message**

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

**GlobalConfiguration**

Global Configuration

**inputLossBehavior**

Settings for system actions when input is lost.

- **Type**: InputLossBehavior (p. 100)
**Required:** False

**supportLowFramerateInputs**

Adjusts video input buffer for streams with very low video framerates. This is commonly set to enabled for music channels with less than one video frame per second.

- **Type:** GlobalConfigurationLowFramerateInputs (p. 74)
- **Required:** False

**outputLockingMode**

Indicates how MediaLive pipelines are synchronized. PIPELINELOCKING - MediaLive will attempt to synchronize the output of each pipeline to the other. EPOCHLOCKING - MediaLive will attempt to synchronize the output of each pipeline to the Unix epoch.

- **Type:** GlobalConfigurationOutputLockingMode (p. 74)
- **Required:** False

**initialAudioGain**

Value to set the initial audio gain for the Live Event.

- **Type:** integer
- **Required:** False
- **Minimum:** -60
- **Maximum:** 60

**inputEndAction**

Indicates the action to take when the current input completes (e.g. end-of-file). When switchAndLoopInputs is configured the encoder will restart at the beginning of the first input. When "none" is configured the encoder will transcode either black, a solid color, or a user specified slate images per the "Input Loss Behavior" configuration until the next input switch occurs (which is controlled through the Channel Schedule API).

- **Type:** GlobalConfigurationInputEndAction (p. 73)
- **Required:** False

**outputTimingSource**

Indicates whether the rate of frames emitted by the Live encoder should be paced by its system clock (which optionally may be locked to another source via NTP) or should be locked to the clock of the source that is providing the input stream.

- **Type:** GlobalConfigurationOutputTimingSource (p. 74)
- **Required:** False

**GlobalConfigurationInputEndAction**

Global Configuration Input End Action

- NONE
- SWITCH_AND_LOOP_INPUTS
GlobalConfigurationLowFramerateInputs
Global Configuration Low Framerate Inputs
   DISABLED
   ENABLED

GlobalConfigurationOutputLockingMode
Global Configuration Output Locking Mode
   EPOCH_LOCKING
   PIPELINE_LOCKING

GlobalConfigurationOutputTimingSource
Global Configuration Output Timing Source
   INPUT_CLOCK
   SYSTEM_CLOCK

H264AdaptiveQuantization
H264 Adaptive Quantization
   HIGH
   HIGHER
   LOW
   MAX
   MEDIUM
   OFF

H264ColorMetadata
H264 Color Metadata
   IGNORE
   INSERT

H264EntropyEncoding
H264 Entropy Encoding
   CABAC
   CAVLC

H264FlickerAq
H264 Flicker Aq
   DISABLED
ENABLED

**H264FramerateControl**

H264 Framerate Control

- INITIALIZE_FROM_SOURCE
- SPECIFIED

**H264GopBReference**

H264 Gop BReference

- DISABLED
- ENABLED

**H264GopSizeUnits**

H264 Gop Size Units

- FRAMES
- SECONDS

**H264Level**

H264 Level

- H264_LEVEL_1
- H264_LEVEL_1_1
- H264_LEVEL_1_2
- H264_LEVEL_1_3
- H264_LEVEL_2
- H264_LEVEL_2_1
- H264_LEVEL_2_2
- H264_LEVEL_3
- H264_LEVEL_3_1
- H264_LEVEL_3_2
- H264_LEVEL_4
- H264_LEVEL_4_1
- H264_LEVEL_4_2
- H264_LEVEL_5
- H264_LEVEL_5_1
- H264_LEVEL_5_2
- H264_LEVEL_AUTO

**H264LookAheadRateControl**

H264 Look Ahead Rate Control

- HIGH
- LOW
- MEDIUM
H264ParControl
H264 Par Control

INITIALIZE_FROM_SOURCE
SPECIFIED

H264Profile
H264 Profile

BASELINE
HIGH
HIGH_10BIT
HIGH_422
HIGH_422_10BIT
MAIN

H264RateControlMode
H264 Rate Control Mode

CBR
QVBR
VBR

H264ScanType
H264 Scan Type

INTERLACED
PROGRESSIVE

H264SceneChangeDetect
H264 Scene Change Detect

DISABLED
ENABLED

H264Settings
H264 Settings

minIInterval

Only meaningful if sceneChangeDetect is set to enabled. 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. Note: Maximum GOP stretch = GOP size + Min-I-interval - 1

Type: integer
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. This field is optional; when no value is specified the encoder will choose the number of slices based on encode resolution.

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

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

Type: H264GopSizeUnits (p. 75)
Required: False

subgopLength
If set to fixed, use gopNumBFrames B-frames per sub-GOP. If set to dynamic, optimize the number of B-frames used for each sub-GOP to improve visual quality.

Type: H264SubGopLength (p. 82)
Required: False

maxBitrate
For QVBR: See the tooltip for Quality level For VBR: Set the maximum bitrate in order to accommodate expected spikes in the complexity of the video.

Type: integer
Required: False
Minimum: 1000

bitrate
Average bitrate in bits/second. Required when the rate control mode is VBR or CBR. Not used for QVBR. In an MS Smooth output group, each output must have a unique value when its bitrate is rounded down to the nearest multiple of 1000.

Type: integer
**bufFillPct**
Percentage of the buffer that should initially be filled (HRD buffer model).

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

**temporalAq**
If set to enabled, adjust quantization within each frame based on temporal variation of content complexity.

- **Type:** H264TemporalAq (p. 83)
- **Required:** False

**afdSignaling**
Indicates that AFD values will be written into the output stream. If afdSignaling is "auto", the system will try to preserve the input AFD value (in cases where multiple AFD values are valid). If set to "fixed", the AFD value will be the value configured in the fixedAfd parameter.

- **Type:** AfdSignaling (p. 34)
- **Required:** False

**timecodeInsertion**
Determines how timecodes should be inserted into the video elementary stream. - 'disabled': Do not include timecodes - 'picTimingSei': Pass through picture timing SEI messages from the source specified in Timecode Config

- **Type:** H264TimecodeInsertionBehavior (p. 83)
- **Required:** False

**bufSize**
Size of buffer (HRD buffer model) in bits/second.

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

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

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

This field indicates how the output video frame rate is specified. If "specified" is selected then the output video frame rate is determined by framerateNumerator and framerateDenominator, else if "initializeFromSource" is selected then the output video frame rate will be set equal to the input video frame rate of the first input.

- **Type**: H264FramerateControl (p. 75)
- **Required**: False

qvbrQualityLevel

Controls the target quality for the video encode. Applies only when the rate control mode is QVBR. Set values for the QVBR quality level field and Max bitrate field that suit your most important viewing devices. Recommended values are:
- Primary screen: Quality level: 8 to 10. Max bitrate: 4M
- PC or tablet: Quality level: 7. Max bitrate: 1.5M to 3M
- Smartphone: Quality level: 6. Max bitrate: 1M to 1.5M

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 10

fixedAfd

Four bit AFD value to write on all frames of video in the output stream. Only valid when afdSignaling is set to 'Fixed'.

- **Type**: FixedAfd (p. 71)
- **Required**: False

level

H.264 Level.

- **Type**: H264Level (p. 75)
- **Required**: False

lookAheadRateControl

Amount of lookahead. A value of low can decrease latency and memory usage, while high can produce better quality for certain content.

- **Type**: H264LookAheadRateControl (p. 75)
- **Required**: False

profile

H.264 Profile.

- **Type**: H264Profile (p. 76)
- **Required**: False

framerateNumerator

Framerate numerator - framerate is a fraction, e.g. 24000 / 1001 = 23.976 fps.
Type: integer
Required: False
Minimum: 1

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

entropyEncoding

Entropy encoding mode. Use cabac (must be in Main or High profile) or cavlc.

Type: H264EntropyEncoding (p. 74)
Required: False

framerateDenominator

Framerate denominator.

Type: integer
Required: False
Minimum: 1

spatialAq

If set to enabled, adjust quantization within each frame based on spatial variation of content complexity.

Type: H264SpatialAq (p. 82)
Required: False

adaptiveQuantization

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

Type: H264AdaptiveQuantization (p. 74)
Required: False

colorMetadata

Includes colorspace metadata in the output.

Type: H264ColorMetadata (p. 74)
Required: False

gopSize

GOP size (keyframe interval) in units of either frames or seconds per gopSizeUnits. Must be greater than zero.
**Properties**

**Type**: number
**Required**: False
**Minimum**: 1.0

**numRefFrames**

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

**gopBReference**

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

**Type**: H264GopBReference (p. 75)
**Required**: False

**parControl**

This field indicates how the output pixel aspect ratio is specified. If "specified" is selected then the output video pixel aspect ratio is determined by parNumerator and parDenominator, else if "initializeFromSource" is selected then the output pixel aspect ratio will be set equal to the input video pixel aspect ratio of the first input.

**Type**: H264ParControl (p. 76)
**Required**: False

**parDenominator**

Pixel Aspect Ratio denominator.

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

**syntax**

Produces a bitstream compliant with SMPTE RP-2027.

**Type**: H264Syntax (p. 82)
**Required**: False

**sceneChangeDetect**

Scene change detection. - On: inserts I-frames when scene change is detected. - Off: does not force an I-frame when scene change is detected.

**Type**: H264SceneChangeDetect (p. 76)
**Required**: False
scanType
Sets the scan type of the output to progressive or top-field-first interlaced.

Type: H264ScanType (p. 76)
Required: False

flickerAq
If set to enabled, adjust quantization within each frame to reduce flicker or ‘pop’ on I-frames.

Type: H264FlickerAq (p. 74)
Required: False

gopNumBFrames
Number of B-frames between reference frames.

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

rateControlMode
Rate control mode. QVBR: Quality will match the specified quality level except when it is constrained by the maximum bitrate. Recommended if you or your viewers pay for bandwidth. VBR: Quality and bitrate vary, depending on the video complexity. Recommended instead of QVBR if you want to maintain a specific average bitrate over the duration of the channel. CBR: Quality varies, depending on the video complexity. Recommended only if you distribute your assets to devices that cannot handle variable bitrates.

Type: H264RateControlMode (p. 76)
Required: False

H264SpatialAq
H264 Spatial Aq
DISABLED
ENABLED

H264SubGopLength
H264 Sub Gop Length
DYNAMIC
FIXED

H264Syntax
H264 Syntax
DEFAULT
RP2027

**H264TemporalAq**

H264 Temporal Aq

- DISABLED
- ENABLED

**H264TimecodeInsertionBehavior**

H264 Timecode Insertion Behavior

- DISABLED
- PIC_TIMING_SEI

**HlsAdMarkers**

Hls Ad Markers

- ADOBE
- ELEMENTAL
- ELEMENTAL_SCTE35

**HlsAkamaiHttpTransferMode**

Hls Akamai Http Transfer Mode

- CHUNKED
- NON_CHUNKED

**HlsAkamaiSettings**

Hls Akamai Settings

salt

Salt for authenticated Akamai.

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

httpTransferMode

Specify whether or not to use chunked transfer encoding to Akamai. User should contact Akamai to enable this feature.

- **Type:** HlsAkamaiHttpTransferMode (p. 83)
- **Required:** False

numRetries

Number of retry attempts that will be made before the Live Event is put into an error state.
**Properties**

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

**restartDelay**

If a streaming output fails, number of seconds to wait until a restart is initiated. A value of 0 means never restart.

```plaintext
Type: integer  
Required: False  
Minimum: 0  
Maximum: 15
```

**connectionRetryInterval**

Number of seconds to wait before retrying connection to the CDN if the connection is lost.

```plaintext
Type: integer  
Required: False  
Minimum: 0
```

**filecacheDuration**

Size in seconds of file cache for streaming outputs.

```plaintext
Type: integer  
Required: False  
Minimum: 0  
Maximum: 600
```

**token**

Token parameter for authenticated akamai. If not specified, _gda_ is used.

```plaintext
Type: string  
Required: False
```

**HlsBasicPutSettings**

Hls Basic Put Settings

**numRetries**

Number of retry attempts that will be made before the Live Event is put into an error state.

```plaintext
Type: integer  
Required: False  
Minimum: 0
```

**restartDelay**

If a streaming output fails, number of seconds to wait until a restart is initiated. A value of 0 means never restart.
**Properties**

**connectionRetryInterval**

Number of seconds to wait before retrying connection to the CDN if the connection is lost.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 15

**filecacheDuration**

Size in seconds of file cache for streaming outputs.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 600

**HlsCaptionLanguageSetting**

Hls Caption Language Setting

- INSERT
- NONE
- OMIT

**HlsCdnSettings**

Hls Cdn Settings

**hlsAkamaiSettings**

- **Type**: HlsAkamaiSettings (p. 83)
- **Required**: False

**hlsWebdavSettings**

- **Type**: HlsWebdavSettings (p. 96)
- **Required**: False

**hlsBasicPutSettings**

- **Type**: HlsBasicPutSettings (p. 84)
- **Required**: False

**hlsMediaStoreSettings**

- **Type**: HlsMediaStoreSettings (p. 93)
Properties

**Required:** False

**HlsClientCache**

Hls Client Cache

- DISABLED
- ENABLED

**HlsCodecSpecification**

Hls Codec Specification

- RFC_4281
- RFC_6381

**HlsDirectoryStructure**

Hls Directory Structure

- SINGLE_DIRECTORY
- SUBDIRECTORY_PER_STREAM

**HlsEncryptionType**

Hls Encryption Type

- AES128
- SAMPLE_AES

**HlsGroupSettings**

Hls Group Settings

**segmentsPerSubdirectory**

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

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

**ivInManifest**

For use with encryptionType. The IV (Initialization Vector) is a 128-bit number used in conjunction with the key for encrypting blocks. If set to "include", IV is listed in the manifest, otherwise the IV is not in the manifest.

- **Type:** HlsIvInManifest (p. 93)
- **Required:** False
outputSelection
MANIFESTSANDSEGMENTS: Generates manifests (master manifest, if applicable, and media manifests) for this output group. SEGMENTSONLY: Does not generate any manifests for this output group.

Type: HlsOutputSelection (p. 94)
Required: False

destination
A directory or HTTP destination for the HLS segments, manifest files, and encryption keys (if enabled).

Type: OutputLocationRef (p. 125)
Required: True

encryptionType
Encrypts the segments with the given encryption scheme. Exclude this parameter if no encryption is desired.

Type: HlsEncryptionType (p. 86)
Required: False

indexNSegments
Applies only if Mode field is LIVE. Specifies the maximum number of segments in the media manifest file. After this maximum, older segments are removed from the media manifest. This number must be less than or equal to the Keep Segments field.

Type: integer
Required: False
Minimum: 3

constantIv
For use with encryptionType. This is a 128-bit, 16-byte hex value represented by a 32-character text string. If ivSource is set to "explicit" then this parameter is required and is used as the IV for encryption.

Type: string
Required: False
MinLength: 32
MaxLength: 32

timedMetadataId3Frame
Indicates ID3 frame that has the timecode.

Type: HlsTimedMetadataId3Frame (p. 96)
Required: False

baseUrlManifest
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
Properties

Required: False

captionLanguageSetting
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.

Type: HlsCaptionLanguageSetting (p. 85)
Required: False

minSegmentLength
When set, minimumSegmentLength 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: False
Minimum: 0

mode
If "vod", all segments are indexed and kept permanently in the destination and manifest. If "live", only the number segments specified in keepSegments and indexNSegments are kept; newer segments replace older segments, which may prevent players from rewinding all the way to the beginning of the event. VOD mode uses HLS EXT-X-PLAYLIST-TYPE of EVENT while the channel is running, converting it to a "VOD" type manifest on completion of the stream.

Type: HlsMode (p. 94)
Required: False

keyProviderSettings
The key provider settings.

Type: KeyProviderSettings (p. 103)
Required: False

manifestCompression
When set to gzip, compresses HLS playlist.

Type: HlsManifestCompression (p. 93)
Required: False

ivSource
For use with encryptionType. The IV (Initialization Vector) is a 128-bit number used in conjunction with the key for encrypting blocks. If this setting is "followsSegmentNumber", it will cause the IV to change every segment (to match the segment number). If this is set to "explicit", you must enter a constant IV value.
**Properties**

- **Type**: `HlsIvSource (p. 93)`
  - **Required**: False

**tsFileMode**

SEGMENTEDFILES: Emit the program as segments - multiple .ts media files. SINGLEFILE: Applies only if Mode field is VOD. Emit the program as a single .ts media file. The media manifest includes #EXT-X-BYTERANGE tags to index segments for playback. A typical use for this value is when sending the output to AWS Elemental MediaConvert, which can accept only a single media file. Playback while the channel is running is not guaranteed due to HTTP server caching.

- **Type**: `HlsTsFileMode (p. 96)`
  - **Required**: False

**manifestDurationFormat**

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

- **Type**: `HlsManifestDurationFormat (p. 93)`
  - **Required**: False

**keyFormatVersions**

Either a single positive integer version value or a slash delimited list of version values (1/2/3).

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

**streamInfResolution**

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

- **Type**: `HlsStreamInfResolution (p. 96)`
  - **Required**: False

**timestampDeltaMilliseconds**

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

- **Type**: `integer`
  - **Required**: False
  - **Minimum**: 0

**baseUrlContent**

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

**segmentationMode**

useInputSegmentation has been deprecated. The configured segment size is always used.
Type: HlsSegmentationMode (p. 95)
Required: False

captionLanguageMappings

Mapping of up to 4 caption channels to caption languages. Is only meaningful if captionLanguageSetting is set to "insert".

Type: Array of type CaptionLanguageMapping (p. 49)
Required: False

clientCache

When set to "disabled", sets the #EXT-X-ALLOW-CACHE:no tag in the manifest, which prevents clients from saving media segments for later replay.

Type: HlsClientCache (p. 86)
Required: False

codecSpecification

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

Type: HlsCodecSpecification (p. 86)
Required: False

keepSegments

Applies only if Mode field is LIVE. Specifies the number of media segments (.ts files) to retain in the destination directory.

Type: integer
Required: False
Minimum: 1

redundantManifest

ENABLED: The master manifest (.m3u8 file) for each pipeline includes information about both pipelines: first its own media files, then the media files of the other pipeline. This feature allows playout device that support stale manifest detection to switch from one manifest to the other, when the current manifest seems to be stale. There are still two destinations and two master manifests, but both master manifests reference the media files from both pipelines. DISABLED: The master manifest (.m3u8 file) for each pipeline includes information about its own pipeline only. For an HLS output group with MediaPackage as the destination, the DISABLED behavior is always followed. MediaPackage regenerates the manifests it serves to players so a redundant manifest from MediaLive is irrelevant.

Type: HlsRedundantManifest (p. 95)
Required: False

timedMetadataId3Period

Timed Metadata interval in seconds.

Type: integer
programDateTime

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 timestampOffset.

Type: HlsProgramDateTime (p. 95)
Required: False

directoryStructure

Place segments in subdirectories.

Type: HlsDirectoryStructure (p. 86)
Required: False

keyFormat

The value specifies how the key is represented in the resource identified by the URI. If parameter is absent, an implicit value of "identity" is used. A reverse DNS string can also be given.

Type: string
Required: False

inputLossAction

Parameter that control output group behavior on input loss.

Type: InputLossActionForHlsOut (p. 99)
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 HlsAdMarkers (p. 83)
Required: False

programDateTimePeriod

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

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

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.
Properties

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

### hlsCdnSettings

Parameters that control interactions with the CDN.

**Type**: HlsCdnSettings (p. 85)  
**Required**: False

### iFrameOnlyPlaylists

DISABLED: Do not create an I-frame-only manifest, but do create the master and media manifests (according to the Output Selection field). STANDARD: Create an I-frame-only manifest for each output that contains video, as well as the other manifests (according to the Output Selection field). The I-frame manifest contains a #EXT-X-I-FRAMES-ONLY tag to indicate it is I-frame only, and one or more #EXT-X-BYTERANGE entries identifying the I-frame position. For example, #EXT-X-BYTERANGE:160364@1461888"

**Type**: IFrameOnlyPlaylistType (p. 97)  
**Required**: False

### HlsInputSettings

Hls Input Settings

**retries**

The number of consecutive times that attempts to read a manifest or segment must fail before the input is considered unavailable.

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

**bandwidth**

When specified the HLS stream with the m3u8 BANDWIDTH that most closely matches this value will be chosen, otherwise the highest bandwidth stream in the m3u8 will be chosen. The bitrate is specified in bits per second, as in an HLS manifest.

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

**retryInterval**

The number of seconds between retries when an attempt to read a manifest or segment fails.

**Type**: integer  
**Required**: False  
**Minimum**: 0
bufferSegments
When specified, reading of the HLS input will begin this many buffer segments from the end (most recently written segment). When not specified, the HLS input will begin with the first segment specified in the m3u8.

  Type: integer
  Required: False
  Minimum: 0

HlsIvInManifest
Hls Iv In Manifest

  EXCLUDE
  INCLUDE

HlsIvSource
Hls Iv Source

  EXPLICIT
  Follows_SEGMENT_NUMBER

HlsManifestCompression
Hls Manifest Compression

  GZIP
  NONE

HlsManifestDurationFormat
Hls Manifest Duration Format

  FLOATING_POINT
  INTEGER

HlsMediaStoreSettings
Hls Media Store Settings

mediaStoreStorageClass
When set to temporal, output files are stored in non-persistent memory for faster reading and writing.

  Type: HlsMediaStoreStorageClass (p. 94)
  Required: False

numRetries
Number of retry attempts that will be made before the Live Event is put into an error state.

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

**restartDelay**
If a streaming output fails, number of seconds to wait until a restart is initiated. A value of 0 means never restart.

**Type**: integer
**Required**: False
**Minimum**: 0
**Maximum**: 15

**connectionRetryInterval**
Number of seconds to wait before retrying connection to the CDN if the connection is lost.

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

**filecacheDuration**
Size in seconds of file cache for streaming outputs.

**Type**: integer
**Required**: False
**Minimum**: 0
**Maximum**: 600

**HlsMediaStoreStorageClass**
Hls Media Store Storage Class

- TEMPORAL

**HlsMode**
Hls Mode

- LIVE
- VOD

**HlsOutputSelection**
Hls Output Selection

- MANIFESTS_AND_SEGMENTS
- SEGMENTS_ONLY

**HlsOutputSettings**
Hls Output Settings
**segmentModifier**

String concatenated to end of segment filenames.

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

**hlsSettings**

Settings regarding the underlying stream. These settings are different for audio-only outputs.

- **Type**: HlsSettings (p. 95)
- **Required**: True

**nameModifier**

String concatenated to the end of the destination filename. Accepts "Format Identifiers \\"#formatIdentifierParameters.

- **Type**: string
- **Required**: False
- **MinLength**: 1

**HlsProgramDateTime**

Hls Program Date Time

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

**HlsRedundantManifest**

Hls Redundant Manifest

- **DISABLED**
- **ENABLED**

**HlsSegmentationMode**

Hls Segmentation Mode

- **USE_INPUT_SEGMENTATION**
- **USE_SEGMENT_DURATION**

**HlsSettings**

Hls Settings

**standardHlsSettings**

- **Type**: StandardHlsSettings (p. 133)
- **Required**: False
**audioOnlyHlsSettings**

*Type: AudioOnlyHlsSettings (p. 39)*

*Required: False*

**HlsStreamInfResolution**

Hls Stream Inf Resolution

- EXCLUDE
- INCLUDE

**HlsTimedMetadataId3Frame**

Hls Timed Metadata Id3 Frame

- NONE
- PRIV
- TDRL

**HlsTsFileMode**

Hls Ts File Mode

- SEGMENTED_FILES
- SINGLE_FILE

**HlsWebdavHttpTransferMode**

Hls Webdav Http Transfer Mode

- CHUNKED
- NON_CHUNKED

**HlsWebdavSettings**

Hls Webdav Settings

**httpTransferMode**

Specify whether or not to use chunked transfer encoding to WebDAV.

*Type: HlsWebdavHttpTransferMode (p. 96)*

*Required: False*

**numRetries**

Number of retry attempts that will be made before the Live Event is put into an error state.

*Type: integer*

*Required: False*
Properties

Minimum: 0

restartDelay

If a streaming output fails, number of seconds to wait until a restart is initiated. A value of 0 means never restart.

Type: integer
Required: False
Minimum: 0
Maximum: 15

connectionRetryInterval

Number of seconds to wait before retrying connection to the CDN if the connection is lost.

Type: integer
Required: False
Minimum: 0

filecacheDuration

Size in seconds of file cache for streaming outputs.

Type: integer
Required: False
Minimum: 0
Maximum: 600

IFrameOnlyPlaylistType

When set to "standard", an I-Frame only playlist will be written out for each video output in the output group. This I-Frame only playlist will contain byte range offsets pointing to the I-frame(s) in each segment.

DISABLED
STANDARD

InputAttachment

inputId

The ID of the input

Type: string
Required: False

inputAttachmentName

User-specified name for the attachment. This is required if the user wants to use this input in an input switch action.

Type: string
**inputSettings**
Settings of an input (caption selector, etc.)

*Type: InputSettings (p. 101)*
*Required: False*

**InputChannelLevel**
Input Channel Level

**inputChannel**
The index of the input channel used as a source.

*Type: integer*
*Required: True*
*Minimum: 0*
*Maximum: 15*

**gain**
Remixing value. Units are in dB and acceptable values are within the range from -60 (mute) and 6 dB.

*Type: integer*
*Required: True*
*Minimum: -60*
*Maximum: 6*

**InputCodec**
codec in increasing order of complexity

- MPEG2
- AVC
- HEVC

**InputDeblockFilter**
Input Deblock Filter

- DISABLED
- ENABLED

**InputDenoiseFilter**
Input Denoise Filter

- DISABLED
**ENABLED**

**InputFilter**

Input Filter

- AUTO
- DISABLED
- FORCED

**InputLocation**

Input Location

**passwordParam**

Key used to extract the password from EC2 Parameter store

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

**uri**

Uniform Resource Identifier - This should be a path to a file accessible to the Live system (eg. a http:// URI) depending on the output type. For example, a RTMP destination should have a uri similar to: "rtmp://fmsserver/live".

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

**username**

Username if credentials are required to access a file or publishing point. This can be either a plaintext username, or a reference to an AWS parameter store name from which the username can be retrieved. AWS Parameter store format: "ssm://<parameter name>"

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

**InputLossActionForHlsOut**

Input Loss Action For Hls Out

- EMIT_OUTPUT
- PAUSE_OUTPUT

**InputLossActionForMsSmoothOut**

Input Loss Action For Ms Smooth Out

- EMIT_OUTPUT
- PAUSE_OUTPUT
InputLossActionForRtmpOut

Input Loss Action For Rtmp Out

- EMIT_OUTPUT
- PAUSE_OUTPUT

InputLossActionForUdpOut

Input Loss Action For Udp Out

- DROP_PROGRAM
- DROP_TS
- EMIT_PROGRAM

InputLossBehavior

Input Loss Behavior

inputLossImageType

Indicates whether to substitute a solid color or a slate into the output after input loss exceeds \( \text{blackFrameMsec} \).

Type: InputLossImageType (p. 101)
Required: False

inputLossImageColor

When input loss image type is "color" this field specifies the color to use. Value: 6 hex characters representing the values of RGB.

Type: string
Required: False
MinLength: 6
MaxLength: 6

inputLossImageSlate

When input loss image type is "slate" these fields specify the parameters for accessing the slate.

Type: InputLocation (p. 99)
Required: False

repeatFrameMsec

On input loss, the number of milliseconds to repeat the previous picture before substituting black into the output. A value \( x \), where \( 0 \leq x \leq 1,000,000 \) and a value of 1,000,000 will be interpreted as infinite.

Type: integer
Required: False
Minimum: 0
**Maximum**: 1,000,000

**blackFrameMsec**
On input loss, the number of milliseconds to substitute black into the output before switching to the frame specified by inputLossImageType. A value \( x \), where \( 0 \leq x \leq 1,000,000 \) and a value of 1,000,000 will be interpreted as infinite.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 1,000,000

**InputLossImageType**
Input Loss Image Type

- COLOR
- SLATE

**InputMaximumBitrate**
Maximum input bitrate in megabits per second. Bitrates up to 50 Mbps are supported currently.

- MAX_10_MBPS
- MAX_20_MBPS
- MAX_50_MBPS

**InputResolution**
Input resolution based on lines of vertical resolution in the input; SD is less than 720 lines, HD is 720 to 1080 lines, UHD is greater than 1080 lines

- SD
- HD
- UHD

**InputSettings**
Live Event input parameters. There can be multiple inputs in a single Live Event.

**sourceEndBehavior**
Loop input if it is a file. This allows a file input to be streamed indefinitely.

- **Type**: InputSourceEndBehavior (p. 103)
- **Required**: False

**deblockFilter**
Enable or disable the deblock filter when filtering.

- **Type**: InputDeblockFilter (p. 98)
Required: False

**audioSelectors**

Used to select the audio stream to decode for inputs that have multiple available.

- **Type**: Array of type AudioSelector (p. 40)
- **Required**: False

**networkInputSettings**

Input settings.

- **Type**: NetworkInputSettings (p. 121)
- **Required**: False

**inputFilter**

Turns on the filter for this input. MPEG-2 inputs have the deblinking filter enabled by default. 1) auto - filtering will be applied depending on input type/quality 2) disabled - no filtering will be applied to the input 3) forced - filtering will be applied regardless of input type

- **Type**: InputFilter (p. 99)
- **Required**: False

**videoSelector**

Informs which video elementary stream to decode for input types that have multiple available.

- **Type**: VideoSelector (p. 138)
- **Required**: False

**filterStrength**

Adjusts the magnitude of filtering from 1 (minimal) to 5 (strongest).

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 5

**denoiseFilter**

Enable or disable the denoise filter when filtering.

- **Type**: InputDenoiseFilter (p. 98)
- **Required**: False

**captionSelectors**

Used to select the caption input to use for inputs that have multiple available.

- **Type**: Array of type CaptionSelector (p. 50)
- **Required**: False
**InputSourceEndBehavior**

Input Source End Behavior

- CONTINUE
- LOOP

**InputSpecification**

codec

Input codec

- **Type**: InputCodec (p. 98)
- **Required**: False

resolution

Input resolution, categorized coarsely

- **Type**: InputResolution (p. 101)
- **Required**: False

maximumBitrate

Maximum input bitrate, categorized coarsely

- **Type**: InputMaximumBitrate (p. 101)
- **Required**: False

**InternalServiceError**

message

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

**InvalidRequest**

message

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

**KeyProviderSettings**

Key Provider Settings

staticKeySettings

- **Type**: StaticKeySettings (p. 133)
Required: False

**LimitExceeded**

**message**

Type: string
Required: False

**ListChannelsResultModel**

**channels**

Type: Array of type `ChannelSummary` (p. 54)
Required: False

**nextToken**

Type: string
Required: False

**LogLevel**

The log level the user wants for their channel.

- ERROR
- WARNING
- INFO
- DEBUG
- DISABLED

**M2tsAbsentInputAudioBehavior**

M2ts Absent Input Audio Behavior

- DROP
- ENCODE_SILENCE

**M2tsArib**

M2ts Arib

- DISABLED
- ENABLED

**M2tsAribCaptionsPidControl**

M2ts Arib Captions Pid Control

AUTO
USE_CONFIGURED

**M2tsAudioBufferModel**
M2ts Audio Buffer Model
   - ATSC
   - DVB

**M2tsAudioInterval**
M2ts Audio Interval
   - VIDEO_AND_FIXED_INTERVALS
   - VIDEO_INTERVAL

**M2tsAudioStreamType**
M2ts Audio Stream Type
   - ATSC
   - DVB

**M2tsBufferModel**
M2ts Buffer Model
   - MULTIPLEX
   - NONE

**M2tsCcDescriptor**
M2ts Cc Descriptor
   - DISABLED
   - ENABLED

**M2tsEbifControl**
M2ts Ebif Control
   - NONE
   - PASSTHROUGH

**M2tsEbpPlacement**
M2ts Ebp Placement
   - VIDEO_AND_AUDIO_PIDS
   - VIDEO_PID
**M2tsEsRateInPes**

M2ts Es Rate In Pes

- EXCLUDE
- INCLUDE

**M2tsKlv**

M2ts Klv

- NONE
- PASSTHROUGH

**M2tsPcrControl**

M2ts Pcr Control

- CONFIGURED_PCR_PERIOD
- PCR_EVERY_PES_PACKET

**M2tsRateMode**

M2ts Rate Mode

- CBR
- VBR

**M2tsScte35Control**

M2ts Scte35 Control

- NONE
- PASSTHROUGH

**M2tsSegmentationMarkers**

M2ts Segmentation Markers

- EBP
- EBP_LEGACY
- NONE
- PSI_SEGSTART
- RAI_ADAPT
- RAI_SEGSTART

**M2tsSegmentationStyle**

M2ts Segmentation Style
MAINTAIN_CADENCE
RESET_CADENCE

M2tsSettings

M2ts Settings

audioStreamType

When set to atsc, uses stream type = 0x81 for AC3 and stream type = 0x87 for EAC3. When set to dvb, uses stream type = 0x06.

Type: M2tsAudioStreamType (p. 105)
Required: False

ecmPid

This field is unused and deprecated.

Type: string
Required: False

dvbTeletextPid

Packet Identifier (PID) for input source DVB Teletext data to this output. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

Type: string
Required: False

aribCaptionsPidControl

If set to auto, pid number used for ARIB Captions will be auto-selected from unused pids. If set to useConfigured, ARIB Captions will be on the configured pid number.

Type: M2tsAribCaptionsPidControl (p. 104)
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.

Type: integer
Required: False
Minimum: 0

rateMode

When vbr, does not insert null packets into transport stream to fill specified bitrate. The bitrate setting acts as the maximum bitrate when vbr is set.

Type: M2tsRateMode (p. 106)
**Properties**

**segmentationTime**
The length in seconds of each segment. Required unless markers is set to None.

- **Type:** number
- **Required:** False
- **Minimum:** 1.0

**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. Can be entered as decimal or hexadecimal values. Each PID specified must be in the range of 32 (or 0x20).8182 (or 0x1ff6).

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

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

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

**fragmentTime**
The length in seconds of each fragment. Only used with EBP markers.

- **Type:** number
- **Required:** False
- **Minimum:** 0.0

**ebpLookaheadMs**
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

**ebpAudioInterval**
When videoAndFixedIntervals is selected, 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. Only available when EBP Cablelabs segmentation markers are selected. Partitions 1 and 2 will always follow the video interval.
### Properties

**scte35Pid**

Packet Identifier (PID) of the SCTE-35 stream in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

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

**programNum**

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

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

**pmtInterval**

The number of milliseconds between instances of this table in the output transport stream. Valid values are 0, 10..1000.

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

**pcrPeriod**

Maximum time in milliseconds between Program Clock Reference (PCRs) inserted into the transport stream.

- **Type:** integer
  - **Required:** False
  - **Minimum:** 0
  - **Maximum:** 500

**segmentationStyle**

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 "resetCadence" 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 \( \$segmentationTime \) seconds. When a segmentation style of "maintainCadence" 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 \( \$segmentationTime \) seconds. Note that EBP lookahead is a slight exception to this rule.

- **Type:** M2tsSegmentationStyle (p. 106)
### ebif

If set to passthrough, passes any EBIF data from the input source to this output.

- **Type:** M2tsEbifControl (p. 105)
- **Required:** False

### audioBufferModel

When set to dvb, uses DVB buffer model for Dolby Digital audio. When set to atsc, the ATSC model is used.

- **Type:** M2tsAudioBufferModel (p. 105)
- **Required:** False

### dvbNitSettings

Inserts DVB Network Information Table (NIT) at the specified table repetition interval.

- **Type:** DvbNitSettings (p. 57)
- **Required:** False

### absentInputAudioBehavior

When set to drop, output audio streams will be removed from the program if the selected input audio stream is removed from the input. This allows the output audio configuration to dynamically change based on input configuration. If this is set to encodeSilence, all output audio streams will output encoded silence when not connected to an active input stream.

- **Type:** M2tsAbsentInputAudioBehavior (p. 104)
- **Required:** False

### timedMetadataBehavior

When set to passthrough, timed metadata will be passed through from input to output.

- **Type:** M2tsTimedMetadataBehavior (p. 114)
- **Required:** False

### timedMetadataPid

Packet Identifier (PID) of the timed metadata stream in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

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

### pmtPid

Packet Identifier (PID) for the Program Map Table (PMT) in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

- **Type:** string
Properties

**etvSignalPid**
Packet Identifier (PID) for input source ETV Signal data to this output. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20) to 8192 (or 0x1ff6).

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

**bufferModel**
If set to multiplex, use multiplex buffer model for accurate interleaving. Setting to bufferModel to none can lead to lower latency, but low-memory devices may not be able to play back the stream without interruptions.

**Type:** M2tsBufferModel (p. 105)  
**Required:** False

**scte35Control**
Optionally pass SCTE-35 signals from the input source to this output.

**Type:** M2tsScte35Control (p. 106)  
**Required:** False

**ebpPlacement**
Controls placement of EBP on Audio PIDs. If set to videoAndAudioPids, EBP markers will be placed on the video PID and all audio PIDs. If set to videoPid, EBP markers will be placed on only the video PID.

**Type:** M2tsEbpPlacement (p. 105)  
**Required:** False

**arib**
When set to enabled, uses ARIB-compliant field muxing and removes video descriptor.

**Type:** M2tsArbib (p. 104)  
**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  
**Minimum:** 0.0

**dvbSdtSettings**
Inserts DVB Service Description Table (SDT) at the specified table repetition interval.

**Type:** DvbSdtSettings (p. 58)
Properties

**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. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

*Type:* string
*Required:* False

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

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

**pcrControl**
When set to pcrEveryPesPacket, 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.

*Type:* M2tsPcrControl (p. 106)
*Required:* False

**videoPid**
Packet Identifier (PID) of the elementary video stream in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

*Type:* string
*Required:* False

**esRateInPes**
Include or exclude the ES Rate field in the PES header.

*Type:* M2tsEsRateInPes (p. 106)
*Required:* False

**segmentationMarkers**
Inserts segmentation markers at each segmentationTime period. raiSegstart sets the Random Access Indicator bit in the adaptation field. raiAdapt sets the RAI bit and adds the current timecode in the private data bytes. psiSegstart 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. ebpLegacy adds Encoder Boundary Point information to the adaptation field using a legacy proprietary format.

*Type:* M2tsSegmentationMarkers (p. 106)
Properties

Required: False

dvbTdtSettings

Inserts DVB Time and Date Table (TDT) at the specified table repetition interval.

Type: DvbTdtSettings (p. 63)
Required: False

klv

If set to passthrough, passes any KLV data from the input source to this output.

Type: M2tsKlv (p. 106)
Required: False

ccDescriptor

When set to enabled, generates captionServiceDescriptor in PMT.

Type: M2tsCcDescriptor (p. 105)
Required: False

patInterval

The number of milliseconds between instances of this table in the output transport stream. Valid values are 0, 10..1000.

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

etvPlatformPid

Packet Identifier (PID) for input source ETV Platform data to this output. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

Type: string
Required: False

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. Can be entered as decimal or hexadecimal values. Each PID specified must be in the range of 32 (or 0x20)..8182 (or 0x1ff6).

Type: string
Required: False

aribCaptionsPid

Packet Identifier (PID) for ARIB Captions in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).
**Properties**

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

**scte27Pids**

Packet Identifier (PID) for input source SCTE-27 data to this output. Multiple values are accepted, and can be entered in ranges and/or by comma separation. Can be entered as decimal or hexadecimal values. Each PID specified must be in the range of 32 (or 0x20)..8182 (or 0x1ff6).

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

**klvDataPids**

Packet Identifier (PID) for input source KLV data to this output. Multiple values are accepted, and can be entered in ranges and/or by comma separation. Can be entered as decimal or hexadecimal values. Each PID specified must be in the range of 32 (or 0x20)..8182 (or 0x1ff6).

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

**M2tsTimedMetadataBehavior**

M2ts Timed Metadata Behavior

- NO_PASSTHROUGH
- PASSTHROUGH

**M3u8PcrControl**

M3u8 Pcr Control

- CONFIGURED_PCR_PERIOD
- PCR_EVERY_PES_PACKET

**M3u8Scte35Behavior**

M3u8 Scte35 Behavior

- NO_PASSTHROUGH
- PASSTHROUGH

**M3u8Settings**

Settings information for the .m3u8 container

**pmtPid**

Packet Identifier (PID) for the Program Map Table (PMT) in the transport stream. Can be entered as a decimal or hexadecimal value.

**Type**: string
Required: False

dscte35Behavior
If set to passthrough, passes any SCTE-35 signals from the input source to this output.

Type: M3u8Scte35Behavior (p. 114)
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. Can be entered as a decimal or hexadecimal value.

Type: string
Required: False

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. Can be entered as decimal or hexadecimal values.

Type: string
Required: False

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

Type: integer
Required: False
Minimum: 0

scte35Pid
Packet Identifier (PID) of the SCTE-35 stream in the transport stream. Can be entered as a decimal or hexadecimal value.

Type: string
Required: False

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

Type: integer
**Properties**

**Required**: False  
**Minimum**: 0  
**Maximum**: 65535

**pcrControl**

When set to `pcrEveryPesPacket`, 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.

*Type*: `M3u8PcrControl (p. 114)  
**Required**: False

**videoPid**

Packet Identifier (PID) of the elementary video stream in the transport stream. Can be entered as a decimal or hexadecimal value.

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

**pcrPeriod**

Maximum time in milliseconds between Program Clock References (PCRs) inserted into the transport stream.

*Type*: integer  
**Required**: False  
**Minimum**: 0  
**Maximum**: 500

**pmtInterval**

The number of milliseconds between instances of this table in the output transport stream. A value of "0" writes out the PMT once per segment file.

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

**programNum**

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

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

**patInterval**

The number of milliseconds between instances of this table in the output transport stream. A value of "0" writes out the PMT once per segment file.
Properties

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

**timedMetadataPid**
Packet Identifier (PID) of the timed metadata stream in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

Type: string
Required: False

**timedMetadataBehavior**
When set to passthrough, timed metadata is passed through from input to output.

Type: M3u8TimedMetadataBehavior (p. 117)
Required: False

M3u8TimedMetadataBehavior

M3u8 Timed Metadata Behavior

- NO_PASSTHROUGH
- PASSTHROUGH

MediaPackageGroupSettings

Media Package Group Settings

**destination**
MediaPackage channel destination.

Type: OutputLocationRef (p. 125)
Required: True

MediaPackageOutputDestinationSettings

Media Package Output Destination Settings

**channelId**
ID of the channel in MediaPackage that is the destination for this output group. You do not need to specify the individual inputs in MediaPackage; MediaLive will handle the connection of the two MediaLive pipelines to the two MediaPackage inputs. The MediaPackage channel and MediaLive channel must be in the same region.

Type: string
Required: False
MinLength: 1
MediaPackageOutputSettings
Media Package Output Settings

Mp2CodingMode
Mp2 Coding Mode

CODING_MODE_1_0
CODING_MODE_2_0

Mp2Settings
Mp2 Settings

codingMode
The MPEG2 Audio coding mode. Valid values are codingMode10 (for mono) or codingMode20 (for stereo).

Type: Mp2CodingMode (p. 118)
Required: False

bitrate
Average bitrate in bits/second.

Type: number
Required: False

sampleRate
Sample rate in Hz.

Type: number
Required: False

MsSmoothGroupSettings
Ms Smooth Group Settings

fragmentLength
Length of mp4 fragments to generate (in seconds). Fragment length must be compatible with GOP size and framerate.

Type: integer
Required: False
Minimum: 1

eventId
MS Smooth event ID to be sent to the IIS server. Should only be specified if eventIdMode is set to useConfigured.
Properties

Type: string
Required: False

timestampOffset
Timestamp offset for the event. Only used if timestampOffsetMode is set to useConfiguredOffset.

Type: string
Required: False

segmentationMode
useInputSegmentation has been deprecated. The configured segment size is always used.

Type: SmoothGroupSegmentationMode (p. 132)
Required: False

numRetries
Number of retry attempts.

Type: integer
Required: False
Minimum: 0

eventStopBehavior
When set to sendEos, send EOS signal to IIS server when stopping the event

Type: SmoothGroupEventStopBehavior (p. 132)
Required: False

acquisitionPointId
The value of the "Acquisition Point Identity" element used in each message placed in the sparse track. Only enabled if sparseTrackType is not "none".

Type: string
Required: False

sparseTrackType
If set to scte35, use incoming SCTE-35 messages to generate a sparse track in this group of MS-Smooth outputs.

Type: SmoothGroupSparseTrackType (p. 132)
Required: False

timestampOffsetMode
Type of timestamp date offset to use. - useEventStartDate: Use the date the event was started as the offset - useConfiguredOffset: Use an explicitly configured date as the offset

Type: SmoothGroupTimestampOffsetMode (p. 132)
**Required**: False

**destination**

Smooth Streaming publish point on an IIS server. Elemental Live acts as a "Push" encoder to IIS.

**Type**: OutputLocationRef (p. 125)

**Required**: True

**audioOnlyTimecodeControl**

If set to passthrough for an audio-only MS Smooth output, the fragment absolute time will be set to the current timecode. This option does not write timecodes to the audio elementary stream.

**Type**: SmoothGroupAudioOnlyTimecodeControl (p. 131)

**Required**: False

**connectionRetryInterval**

Number of seconds to wait before retrying connection to the IIS server if the connection is lost. Content will be cached during this time and the cache will be be delivered to the IIS server once the connection is re-established.

**Type**: integer

**Required**: False

**Minimum**: 0

**filecacheDuration**

Size in seconds of file cache for streaming outputs.

**Type**: integer

**Required**: False

**Minimum**: 0

**certificateMode**

If set to verifyAuthenticity, verify the https certificate chain to a trusted Certificate Authority (CA). This will cause https outputs to self-signed certificates to fail.

**Type**: SmoothGroupCertificateMode (p. 132)

**Required**: False

**inputLossAction**

Parameter that control output group behavior on input loss.

**Type**: InputLossActionForMsSmoothOut (p. 99)

**Required**: False

**sendDelayMs**

Number of milliseconds to delay the output from the second pipeline.

**Type**: integer
**eventIdMode**

Specifies whether or not to send an event ID to the IIS server. If no event ID is sent and the same Live Event is used without changing the publishing point, clients might see cached video from the previous run. Options: - "useConfigured" - use the value provided in eventId - "useTimestamp" - generate and send an event ID based on the current timestamp - "noEventId" - do not send an event ID to the IIS server.

*Type: SmoothGroupEventIdMode (p. 132)*

**restartDelay**

Number of seconds before initiating a restart due to output failure, due to exhausting the numRetries on one segment, or exceeding filecacheDuration.

*Type: integer*

**streamManifestBehavior**

When set to send, send stream manifest so publishing point doesn't start until all streams start.

*Type: SmoothGroupStreamManifestBehavior (p. 132)*

**MsSmoothOutputSettings**

Ms Smooth Output Settings

**nameModifier**

String concatenated to the end of the destination filename. Required for multiple outputs of the same type.

*Type: string*

**NetworkInputServerValidation**

Network Input Server Validation

- CHECK_CRYPTOGRAPHY_AND_VALIDATE_NAME
- CHECK_CRYPTOGRAPHY_ONLY

**NetworkInputSettings**

Network source to transcode. Must be accessible to the Elemental Live node that is running the live event through a network connection.
**hlsInputSettings**

Specifies HLS input settings when the uri is for a HLS manifest.

- **Type**: HlsInputSettings (p. 92)
- **Required**: False

**serverValidation**

Check HTTPS server certificates. When set to checkCryptographyOnly, cryptography in the certificate will be checked, but not the server's name. Certain subdomains (notably S3 buckets that use dots in the bucket name) do not strictly match the corresponding certificate's wildcard pattern and would otherwise cause the event to error. This setting is ignored for protocols that do not use https.

- **Type**: NetworkInputServerValidation (p. 121)
- **Required**: False

**Output**

Output settings. There can be multiple outputs within a group.

**videoDescriptionName**

The name of the VideoDescription used as the source for this output.

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

**outputName**

The name used to identify an output.

- **Type**: string
- **Required**: False
- **MinLength**: 1
- **MaxLength**: 255

**captionDescriptionNames**

The names of the CaptionDescriptions used as caption sources for this output.

- **Type**: Array of type string
- **Required**: False

**outputSettings**

Output type-specific settings.

- **Type**: OutputSettings (p. 125)
- **Required**: True

**audioDescriptionNames**

The names of the AudioDescriptions used as audio sources for this output.
Properties

Type: Array of type string
Required: False

OutputDestination

mediaPackageSettings

Destination settings for a MediaPackage output; one destination for both encoders.

Type: Array of type MediaPackageOutputDestinationSettings (p. 117)
Required: False

settings

Destination settings for a standard output; one destination for each redundant encoder.

Type: Array of type OutputDestinationSettings (p. 123)
Required: False

id

User-specified id. This is used in an output group or an output.

Type: string
Required: False

OutputDestinationSettings

passwordParam

key used to extract the password from EC2 Parameter store

Type: string
Required: False

streamName

Stream name for RTMP destinations (URLs of type rtmp://)

Type: string
Required: False

url

A URL specifying a destination

Type: string
Required: False

username

username for destination
Properties

Type: string
Required: False

OutputGroup

Output groups for this Live Event. Output groups contain information about where streams should be distributed.

outputs

Type: Array of type Output (p. 122)
Required: True

outputGroupSettings

Settings associated with the output group.

Type: OutputGroupSettings (p. 124)
Required: True

name

Custom output group name optionally defined by the user. Only letters, numbers, and the underscore character allowed; only 32 characters allowed.

Type: string
Required: False
MaxLength: 32

OutputGroupSettings

Output Group Settings

archiveGroupSettings

Type: ArchiveGroupSettings (p. 34)
Required: False

mediaPackageGroupSettings

Type: MediaPackageGroupSettings (p. 117)
Required: False

rtmpGroupSettings

Type: RtmpGroupSettings (p. 127)
Required: False

udpGroupSettings

Type: UdpGroupSettings (p. 135)
Required: False
msSmoothGroupSettings
Type: MsSmoothGroupSettings (p. 118)
Required: False

hlsGroupSettings
Type: HlsGroupSettings (p. 86)
Required: False

frameCaptureGroupSettings
Type: FrameCaptureGroupSettings (p. 71)
Required: False

OutputLocationRef
Reference to an OutputDestination ID defined in the channel
destinationRefId
Type: string
Required: False

OutputSettings
Output Settings
rtmpOutputSettings
Type: RtmpOutputSettings (p. 128)
Required: False

archiveOutputSettings
Type: ArchiveOutputSettings (p. 34)
Required: False

frameCaptureOutputSettings
Type: FrameCaptureOutputSettings (p. 72)
Required: False

msSmoothOutputSettings
Type: MsSmoothOutputSettings (p. 121)
Required: False

mediaPackageOutputSettings
Type: MediaPackageOutputSettings (p. 118)
Required: False
udpOutputSettings

Type: UdpOutputSettings (p. 135)
Required: False

hlsOutputSettings

Type: HlsOutputSettings (p. 94)
Required: False

PassThroughSettings
Pass Through Settings

RemixSettings
Remix Settings

channelMappings
Mapping of input channels to output channels, with appropriate gain adjustments.

Type: Array of type AudioChannelMapping (p. 35)
Required: True

channelsOut
Number of output channels to be produced. Valid values: 1, 2, 4, 6, 8

Type: integer
Required: False
Minimum: 1
Maximum: 8

channelsIn
Number of input channels to be used.

Type: integer
Required: False
Minimum: 1
Maximum: 16

ResourceConflict
message

Type: string
Required: False

RtmpCacheFullBehavior
Rtmp Cache Full Behavior
**DISCONNECT_IMMEDIATELY**
**WAIT_FOR_SERVER**

**RtmpCaptionData**

Rtmp Caption Data

- **ALL**
- **FIELD1_608**
- **FIELD1_AND_FIELD2_608**

**RtmpCaptionInfoDestinationSettings**

Rtmp Caption Info Destination Settings

**RtmpGroupSettings**

Rtmp Group Settings

**inputLossAction**

Controls the behavior of this RTMP group if input becomes unavailable. - emitOutput: Emit a slate until input returns. - pauseOutput: Stop transmitting data until input returns. This does not close the underlying RTMP connection.

- **Type**: InputLossActionForRtmpOut (p. 100)
- **Required**: False

**captionData**

Controls the types of data that passes to onCaptionInfo outputs. If set to 'all' then 608 and 708 carried DTVCC data will be passed. If set to 'field1AndField2608' then DTVCC data will be stripped out, but 608 data from both fields will be passed. If set to 'field1608' then only the data carried in 608 from field 1 video will be passed.

- **Type**: RtmpCaptionData (p. 127)
- **Required**: False

**authenticationScheme**

Authentication scheme to use when connecting with CDN

- **Type**: AuthenticationScheme (p. 41)
- **Required**: False

**cacheLength**

Cache length, in seconds, is used to calculate buffer size.

- **Type**: integer
  - **Required**: False
  - **Minimum**: 30
restartDelay

If a streaming output fails, number of seconds to wait until a restart is initiated. A value of 0 means never restart.

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

cacheFullBehavior

Controls behavior when content cache fills up. If remote origin server stalls the RTMP connection and does not accept content fast enough the 'Media Cache' will fill up. When the cache reaches the duration specified by cacheLength the cache will stop accepting new content. If set to disconnectImmediately, the RTMP output will force a disconnect. Clear the media cache, and reconnect after restartDelay seconds. If set to waitForServer, the RTMP output will wait up to 5 minutes to allow the origin server to begin accepting data again.

- **Type**: RtmpCacheFullBehavior (p. 126)
- **Required**: False

RtmpOutputCertificateMode

Rtmp Output Certificate Mode

- SELF_SIGNED
- VERIFY_AUTHENTICITY

RtmpOutputSettings

Rtmp Output Settings

certificateMode

If set to verifyAuthenticity, verify the tls certificate chain to a trusted Certificate Authority (CA). This will cause rtmps outputs with self-signed certificates to fail.

- **Type**: RtmpOutputCertificateMode (p. 128)
- **Required**: False

numRetries

Number of retry attempts.

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

destination

The RTMP endpoint excluding the stream name (eg. rtmp://host/appname). For connection to Akamai, a username and password must be supplied. URI fields accept format identifiers.

- **Type**: OutputLocationRef (p. 125)
**Required**: True

**connectionRetryInterval**

Number of seconds to wait before retrying a connection to the Flash Media server if the connection is lost.

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

**Scte20Convert608To708**

Scte20 Convert608 To708

- DISABLED
- UPCONVERT

**Scte20PlusEmbeddedDestinationSettings**

Scte20 Plus Embedded Destination Settings

**Scte20SourceSettings**

Scte20 Source Settings

**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**

If 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.

*Type*: [Scte20Convert608To708](p. 129)  
*Required*: False

**Scte27DestinationSettings**

Scte27 Destination Settings

**Scte27SourceSettings**

Scte27 Source Settings
Properties

pid

The pid field is used in conjunction with the caption selector languageCode field as follows: - Specify PID and Language: Extracts captions from that PID; the language is "informational". - Specify PID and omit Language: Extracts the specified PID. - Omit PID and specify Language: Extracts the specified language, whichever PID that happens to be. - Omit PID and omit Language: Valid only if source is DVB-Sub that is being passed through; all languages will be passed through.

  Type: integer
  Required: False
  Minimum: 1

Scte35AposNoRegionalBlackoutBehavior

Scte35 Apos No Regional Blackout Behavior

  FOLLOW
  IGNORE

Scte35AposWebDeliveryAllowedBehavior

Scte35 Apos Web Delivery Allowed Behavior

  FOLLOW
  IGNORE

Scte35SpliceInsert

Scte35 Splice Insert

adAvailOffset

When specified, this offset (in milliseconds) is added to the input Ad Avail PTS time. This only applies to embedded SCTE 104/35 messages and does not apply to OOB messages.

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

webDeliveryAllowedFlag

When set to ignore, Segment Descriptors with webDeliveryAllowedFlag set to 0 will no longer trigger blackouts or Ad Avail slates

  Type: Scte35SpliceInsertWebDeliveryAllowedBehavior (p. 131)
  Required: False

noRegionalBlackoutFlag

When set to ignore, Segment Descriptors with noRegionalBlackoutFlag set to 0 will no longer trigger blackouts or Ad Avail slates
**Type:** `Scte35SpliceInsertNoRegionalBlackoutBehavior` (p. 131)
**Required:** False

*Scte35SpliceInsertNoRegionalBlackoutBehavior*

Scte35 Splice Insert No Regional Blackout Behavior

- FOLLOW
- IGNORE

*Scte35SpliceInsertWebDeliveryAllowedBehavior*

Scte35 Splice Insert Web Delivery Allowed Behavior

- FOLLOW
- IGNORE

*Scte35TimeSignalApos*

Scte35 Time Signal Apos

**adAvailOffset**

When specified, this offset (in milliseconds) is added to the input Ad Avail PTS time. This only applies to embedded SCTE 104/35 messages and does not apply to OOB messages.

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

**webDeliveryAllowedFlag**

When set to ignore, Segment Descriptors with webDeliveryAllowedFlag set to 0 will no longer trigger blackouts or Ad Avail slates

- **Type:** `Scte35AposWebDeliveryAllowedBehavior` (p. 130)
- **Required:** False

**noRegionalBlackoutFlag**

When set to ignore, Segment Descriptors with noRegionalBlackoutFlag set to 0 will no longer trigger blackouts or Ad Avail slates

- **Type:** `Scte35AposNoRegionalBlackoutBehavior` (p. 130)
- **Required:** False

**SmoothGroupAudioOnlyTimecodeControl**

Smooth Group Audio Only Timecode Control

- PASSTHROUGH
USE_CONFIGURED_CLOCK

SmoothGroupCertificateMode
Smooth Group Certificate Mode

SELF_SIGNED
VERIFY_AUTHENTICITY

SmoothGroupEventIdMode
Smooth Group Event Id Mode

NO_EVENT_ID
USE_CONFIGURED
USE_TIMESTAMP

SmoothGroupEventStopBehavior
Smooth Group Event Stop Behavior

NONE
SEND_EOS

SmoothGroupSegmentationMode
Smooth Group Segmentation Mode

USE_INPUT_SEGMENTATION
USE_SEGMENT_DURATION

SmoothGroupSparseTrackType
Smooth Group Sparse Track Type

NONE
SCTE_35

SmoothGroupStreamManifestBehavior
Smooth Group Stream Manifest Behavior

DO_NOT_SEND
SEND

SmoothGroupTimestampOffsetMode
Smooth Group Timestamp Offset Mode

USE_CONFIGURED_OFFSET
USE_EVENT_START_DATE
SmpteTtDestinationSettings

Smpte Tt Destination Settings

StandardHlsSettings

Standard Hls Settings

m3u8Settings

Type: M3u8Settings (p. 114)
Required: True

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

StaticKeySettings

Static Key Settings

staticKeyValue

Static key value as a 32 character hexadecimal string.

Type: string
Required: True
MinLength: 32
MaxLength: 32

keyProviderServer

The URL of the license server used for protecting content.

Type: InputLocation (p. 99)
Required: False

Tags

key-value pairs

Type: string

TeletextDestinationSettings

Teletext Destination Settings

TeletextSourceSettings

Teletext Source Settings
**pageNumber**

Specifies the teletext page number within the data stream from which to extract captions. Range of 0x100 (256) to 0x8FF (2303). Unused for passthrough. Should be specified as a hexadecimal string with no "0x" prefix.

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

**TimecodeConfig**

Timecode Config

**syncThreshold**

Threshold in frames beyond which output timecode is resynchronized to the input timecode. Discrepancies below this threshold are permitted to avoid unnecessary discontinuities in the output timecode. No timecode sync when this is not specified.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 1000000

**source**

Identifies the source for the timecode that will be associated with the events outputs. - Embedded (embedded): Initialize the output timecode with timecode from the the source. If no embedded timecode is detected in the source, the system falls back to using "Start at 0" (zerobased). - System Clock (systemclock): Use the UTC time. - Start at 0 (zerobased): The time of the first frame of the event will be 00:00:00:00.

- **Type:** TimecodeConfigSource (p. 134)
- **Required:** True

**TimecodeConfigSource**

Timecode Config Source

<table>
<thead>
<tr>
<th>Embedded</th>
<th>SystemClock</th>
<th>Zerobased</th>
</tr>
</thead>
</table>

**TtmlDestinationSettings**

Ttml Destination Settings

**styleControl**

When set to passthrough, passes through style and position information from a TTML-like input source (TTML, SMPTE-TT, CFF-TT) to the CFF-TT output or TTML output.

- **Type:** TtmlDestinationStyleControl (p. 135)
- **Required:** False
**TttmlDestinationStyleControl**

Tttml Destination Style Control

- PASSTHROUGH
- USE_CONFIGURED

**UdpContainerSettings**

Udp Container Settings

**m2tsSettings**

- Type: M2tsSettings (p. 107)
- Required: False

**UdpGroupSettings**

Udp Group Settings

**inputLossAction**

Specifies behavior of last resort when input video is lost, and no more backup inputs are available. When dropTs is selected the entire transport stream will stop being emitted. When dropProgram is selected the program can be dropped from the transport stream (and replaced with null packets to meet the TS bitrate requirement). Or, when emitProgram is chosen the transport stream will continue to be produced normally with repeat frames, black frames, or slate frames substituted for the absent input video.

- Type: InputLossActionForUdpOut (p. 100)
- Required: False

**timedMetadataId3Frame**

Indicates ID3 frame that has the timecode.

- Type: UdpTimedMetadataId3Frame (p. 136)
- Required: False

**timedMetadataId3Period**

Timed Metadata interval in seconds.

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

**UdpOutputSettings**

Udp Output Settings

**destination**

Destination address and port number for RTP or UDP packets. Can be unicast or multicast RTP or UDP (eg. rtp://239.10.10.10:50001 or udp://10.100.100.100:5002).
**Properties**

### bufferMsec

UDP output buffering in milliseconds. Larger values increase latency through the transcoder but simultaneously assist the transcoder in maintaining a constant, low-jitter UDP/RTP output while accommodating clock recovery, input switching, input disruptions, picture reordering, etc.

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

### containerSettings

- **Type:** UdpContainerSettings (p. 135)
- **Required:** True

### fecOutputSettings

Settings for enabling and adjusting Forward Error Correction on UDP outputs.

- **Type:** FecOutputSettings (p. 71)
- **Required:** False

### UdpTimedMetadataId3Frame

Udp Timed Metadata Id3 Frame

- **NONE**
- **PRIV**
- **TDRL**

### ValidationException

### errorMessage

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

### elementPath

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

### VideoCodecSettings

Video Codec Settings

### h264Settings

- **Type:** H264Settings (p. 76)
Properties

**Required**: False

**frameCaptureSettings**

*Type*: FrameCaptureSettings (p. 72)

*Required*: False

**VideoDescription**

Video settings for this stream.

**respondToAfd**

Indicates how to respond to the AFD values in the input stream. RESPOND causes input video to be clipped, depending on the AFD value, input display aspect ratio, and output display aspect ratio, and (except for FRAMECAPTURE codec) includes the values in the output. PASSTHROUGH (does not apply to FRAMECAPTURE codec) ignores the AFD values and includes the values in the output, so input video is not clipped. NONE ignores the AFD values and does not include the values through to the output, so input video is not clipped.

*Type*: VideoDescriptionRespondToAfd (p. 138)

*Required*: False

**scalingBehavior**

STRETCHTOOUTPUT configures the output position to stretch the video to the specified output resolution (height and width). This option will override any position value. DEFAULT may insert black boxes (pillar boxes or letter boxes) around the video to provide the specified output resolution.

*Type*: VideoDescriptionScalingBehavior (p. 138)

*Required*: False

**name**

The name of this VideoDescription. Outputs will use this name to uniquely identify this Description. Description names should be unique within this Live Event.

*Type*: string

*Required*: True

**width**

Output video width, in pixels. Must be an even number. For most codecs, you can leave this field and height blank in order to use the height and width (resolution) from the source. Note, however, that leaving blank is not recommended. For the Frame Capture codec, height and width are required.

*Type*: integer

*Required*: False

**sharpness**

Changes the strength of the anti-alias filter used for scaling. 0 is the softest setting, 100 is the sharpest. A setting of 50 is recommended for most content.
Properties

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

**codecSettings**
Video codec settings.

- **Type**: VideoCodecSettings (p. 136)  
- **Required**: False

**height**
Output video height, in pixels. Must be an even number. For most codecs, you can leave this field and width blank in order to use the height and width (resolution) from the source. Note, however, that leaving blank is not recommended. For the Frame Capture codec, height and width are required.

- **Type**: integer  
- **Required**: False

**VideoDescriptionRespondToAfd**
Video Description Respond To Afd

- NONE  
- PASSTHROUGH  
- RESPOND

**VideoDescriptionScalingBehavior**
Video Description Scaling Behavior

- DEFAULT  
- STRETCH_TO_OUTPUT

**VideoSelector**
Specifies a particular video stream within an input source. An input may have only a single video selector.

**colorSpace**
Specifies the colorspace of an input. This setting works in tandem with colorSpaceConversion to determine if any conversion will be performed.

- **Type**: VideoSelectorColorSpace (p. 139)  
- **Required**: False

**selectorSettings**
The video selector settings.

- **Type**: VideoSelectorSettings (p. 140)
**colorSpaceUsage**

Applies only if colorSpace is a value other than follow. This field controls how the value in the colorSpace field will be used. fallback means that when the input does include color space data, that data will be used, but when the input has no color space data, the value in colorSpace will be used. Choose fallback if your input is sometimes missing color space data, but when it does have color space data, that data is correct. force means to always use the value in colorSpace. Choose force if your input usually has no color space data or might have unreliable color space data.

**Type:** VideoSelectorColorSpaceUsage (p. 139)

**Required:** False

**VideoSelectorColorSpace**

Video Selector Color Space

- FOLLOW
- REC_601
- REC_709

**VideoSelectorColorSpaceUsage**

Video Selector Color Space Usage

- FALLBACK
- FORCE

**VideoSelectorPid**

Video Selector Pid

**pid**

Selects a specific PID from within a video source.

**Type:** integer

**Required:** False

**Minimum:** 0

**Maximum:** 8191

**VideoSelectorProgramId**

Video Selector Program Id

**programId**

Selects a specific program from within a multi-program transport stream. If the program doesn't exist, the first program within the transport stream will be selected by default.

**Type:** integer

**Required:** False
Minimum: 0
Maximum: 65536

**VideoSelectorSettings**

Video Selector Settings

**videoSelectorPid**

Type: VideoSelectorPid (p. 139)
Required: False

**videoSelectorProgramId**

Type: VideoSelectorProgramId (p. 139)
Required: False

**WebvttDestinationSettings**

Webvtt Destination Settings

**Channels channelId**

**URI**

/prod/channels/{channelId}

**HTTP Methods**

**GET**

Operation ID: DescribeChannel

Gets details about a channel

**Path Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>channelId</td>
<td>String</td>
<td>True</td>
<td>channel ID</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>Channel (p. 153)</td>
<td>Channel details</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 176)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 176)</td>
<td>You do not have permission to list channels.</td>
</tr>
</tbody>
</table>
AWS Elemental MediaLive API Reference

HTTP Methods

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>404</td>
<td>ResourceNotFound</td>
<td>The channel you're requesting to describe does not exist.</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded</td>
<td>Request limit exceeded on list channel calls to channel service.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>BadGatewayException</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>504</td>
<td>GatewayTimeoutException</td>
<td>Gateway Timeout Error</td>
</tr>
</tbody>
</table>

## PUT

Operation ID: UpdateChannel

Updates a channel.

### Path Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>channelId</td>
<td>String</td>
<td>True</td>
<td>channel ID</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>UpdateChannelResultModel (p. 165)</td>
<td>Channel is successfully updated.</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 176)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 176)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>409</td>
<td>ResourceConflict (p. 176)</td>
<td>The channel is unable to create due to an issue with channel resources.</td>
</tr>
<tr>
<td>422</td>
<td>ChannelConfigurationValidationError (p. 176)</td>
<td>The Channel failed validation and could not be created.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 177)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>BadGatewayException (p. 177)</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>504</td>
<td>GatewayTimeoutException (p. 177)</td>
<td>Gateway Timeout Error</td>
</tr>
</tbody>
</table>

## DELETE

Operation ID: DeleteChannel

Starts deletion of channel. The associated outputs are also deleted.
Path Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>channelId</td>
<td>String</td>
<td>True</td>
<td>channel ID</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>Channel (p. 153)</td>
<td>Deletion was successfully initiated.</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 176)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 176)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>404</td>
<td>ResourceNotFound (p. 176)</td>
<td>The channel you're requesting to describe does not exist.</td>
</tr>
<tr>
<td>409</td>
<td>ResourceConflict (p. 176)</td>
<td>The channel is unable to create due to an issue with channel resources.</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 177)</td>
<td>Request limit exceeded on list channel calls to channel service.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 177)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>BadGatewayException (p. 177)</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>504</td>
<td>GatewayTimeoutException (p. 177)</td>
<td>Gateway Timeout Error</td>
</tr>
</tbody>
</table>

Schemas

Request Bodies

Example PUT

```json
{
  "inputAttachments": [
    {
      "inputId": "string",
      "inputAttachmentName": "string",
      "inputSettings": {
        "sourceEndBehavior": enum,
        "deblockFilter": enum,
        "audioSelectors": [
          {
            "name": "string",
            "selectorSettings": {
              "audioLanguageSelection": {
                "languageSelectionPolicy": enum,
                "languageCode": "string"
              }
            }
          }
        ]
      }
    }
  ]
}
```
"pid": integer
],
"networkInputSettings": {
"hlsInputSettings": {
"retries": integer,
"bandwidth": integer,
"retryInterval": integer,
"bufferSegments": integer
},
"serverValidation": enum
},
"inputFilter": enum,
"videoSelector": {
"colorSpace": enum,
"selectorSettings": {
"videoSelectorPid": {
"pid": integer
},
"videoSelectorProgramId": {
"programId": integer
}
},
"colorSpaceUsage": enum
},
"filterStrength": integer,
"denoiseFilter": enum,
"captionSelectors": [
{
"name": "string",
"languageCode": "string",
"selectorSettings": {
"embeddedSourceSettings": {
"scte20Detection": enum,
"source608ChannelNumber": integer,
"convert608To708": enum,
"source608TrackNumber": integer
},
"scte20SourceSettings": {
"source608ChannelNumber": integer,
"convert608To708": enum
},
"dvbSubSourceSettings": {
"pid": integer
},
"teletextSourceSettings": {
"pageNumber": "string"
},
"aribSourceSettings": {
},
"scte27SourceSettings": {
"pid": integer
}
}]
},
"logLevel": enum,
"roleArn": "string",
"destinations": [
{
"mediaPackageSettings": [
]"networkInputSettings": {
"hlsInputSettings": {
"retries": integer,
"bandwidth": integer,
"retryInterval": integer,
"bufferSegments": integer
},
"serverValidation": enum
},
"inputFilter": enum,
"videoSelector": {
"colorSpace": enum,
"selectorSettings": {
"videoSelectorPid": {
"pid": integer
},
"videoSelectorProgramId": {
"programId": integer
}
},
"colorSpaceUsage": enum
},
"filterStrength": integer,
"denoiseFilter": enum,
"captionSelectors": [
{
"name": "string",
"languageCode": "string",
"selectorSettings": {
"embeddedSourceSettings": {
"scte20Detection": enum,
"source608ChannelNumber": integer,
"convert608To708": enum,
"source608TrackNumber": integer
},
"scte20SourceSettings": {
"source608ChannelNumber": integer,
"convert608To708": enum
},
"dvbSubSourceSettings": {
"pid": integer
},
"teletextSourceSettings": {
"pageNumber": "string"
},
"aribSourceSettings": {
},
"scte27SourceSettings": {
"pid": integer
}
}]
},
"logLevel": enum,
"roleArn": "string",
"destinations": [
{
"mediaPackageSettings": [
]
{
  "channelId": "string"
},
"settings": [
  {
    "passwordParam": "string",
    "streamName": "string",
    "url": "string",
    "username": "string"
  }
],
"id": "string"
},
"name": "string",
"encoderSettings": {
  "timecodeConfig": {
    "syncThreshold": integer,
    "source": enum
  },
  "outputGroups": [
    {
      "outputs": [
        {
          "videoDescriptionName": "string",
          "outputName": "string",
          "captionDescriptionNames": [
            "string"
          ],
          "outputSettings": {
            "rtmpOutputSettings": {
              "certificateMode": enum,
              "numRetries": integer,
              "destination": {
                "destinationRefId": "string"
              },
              "connectionRetryInterval": integer
            },
            "archiveOutputSettings": {
              "extension": "string",
              "containerSettings": {
                "m2tsSettings": {
                  "audioStreamType": enum,
                  "ecmPid": "string",
                  "dvbTeletextPId": "string",
                  "aribCaptionsPidControl": enum,
                  "bitrate": integer,
                  "rateMode": enum,
                  "segmentationTime": number,
                  "audioPids": "string",
                  "audioFramesPerPes": integer,
                  "fragmentTime": number,
                  "ebpLookaheadMs": integer,
                  "ebpAudioInterval": enum,
                  "scte35Pid": "string",
                  "programNum": integer,
                  "pmtInterval": integer,
                  "pcrPeriod": integer,
                  "segmentationStyle": enum,
                  "ebif": enum,
                  "audioBufferModel": enum,
                  "dvbNitSettings": {
                    "networkName": "string",
                    "networkId": integer,
                    "repInterval": integer
                  }
                }
              }
            }
          }
        }
      }
    }
  ]
}
"absentInputAudioBehavior": enum,
"timedMetadataBehavior": enum,
"pmtPid": "string",
"etvSignalPid": "string",
"bufferModel": enum,
"scte35Control": enum,
"ebpPlacement": enum,
"arib": enum,
"nullPacketBitrate": number,
"dvbSdtSettings": {
  "serviceName": "string",
  "serviceProviderName": "string",
  "repInterval": integer,
  "outputSdt": enum
},
"pcrPid": "string",
"transportStreamId": integer,
"pcrControl": enum,
"videoPid": "string",
"esRateInPes": enum,
"segmentationMarkers": enum,
"dvbTdtSettings": {
  "repInterval": integer
},
"klv": enum,
"ccDescriptor": enum,
"patInterval": integer,
"etvPlatformPid": "string",
"dvbSubPids": "string",
"aribCaptionsPid": "string",
"scte27Pids": "string",
"klvDataPids": "string"

"nameModifier": "string"
},
"frameCaptureOutputSettings": {
  "nameModifier": "string"
},
"msSmoothOutputSettings": {
  "nameModifier": "string"
},
"mediaPackageOutputSettings": {
},
"udpOutputSettings": {
  "destination": {
    "destinationRefId": "string"
  },
  "bufferMsec": integer,
  "containerSettings": {
    "m2tsSettings": {
      "audioStreamType": enum,
      "ecmPid": "string",
      "dvbTeletextPid": "string",
      "aribCaptionsPidControl": enum,
      "bitrate": integer,
      "rateMode": enum,
      "segmentationTime": number,
      "audioPids": "string",
      "audioFramesPerPes": integer,
      "fragmentTime": number,
      "ebpLookaheadMs": integer,
      "ebpAudioInterval": enum,
      "scte35Pid": "string",
    }
  }
}
"programNum": integer,
"pmtInterval": integer,
"pcrPeriod": integer,
"segmentationStyle": enum,
"ebif": enum,
"audioBufferModel": enum,
"dvbNitSettings": {
  "networkName": "string",
  "networkId": integer,
  "repInterval": integer
},
"absentInputAudioBehavior": enum,
"timedMetadataBehavior": enum,
"timedMetadataPid": "string",
"pmtPid": "string",
"etvSignalPid": "string",
"bufferModel": enum,
"scte35Control": enum,
"ebpPlacement": enum,
"arib": enum,
"nullPacketBitrate": number,
"dvbSdtSettings": {
  "serviceName": "string",
  "serviceProviderName": "string",
  "repInterval": integer,
  "outputSdt": enum
},
"pcrPid": "string",
"transportStreamId": integer,
"pcrControl": enum,
"videoPid": "string",
"esRateInPes": enum,
"segmentationMarkers": enum,
"dvbTdtSettings": {
  "repInterval": integer
},
"klv": enum,
"ccDescriptor": enum,
"patInterval": integer,
"etvPlatformPid": "string",
"dvbSubPids": "string",
"aribCaptionsPid": "string",
"scte27Pids": "string",
"klvDataPids": "string"
},
"fecOutputSettings": {
  "rowLength": integer,
  "columnDepth": integer,
  "includeFec": enum
},
"hlsOutputSettings": {
  "SegmentModifier": "string",
  "hlsSettings": {
    "standardHlsSettings": {
      "m3u8Settings": {
        "pmtPid": "string",
        "ecmPid": "string",
        "scte35Behavior": enum,
        "pcrPid": "string",
        "audioPids": "string",
        "audioFramesPerPes": integer,
        "scte35Pid": "string",
        "transportStreamId": integer,
        "pcrControl": enum
      }
    }
  }
}
"videoPid": "string",
"pcrPeriod": integer,
"pmtInterval": integer,
"programNum": integer,
"patInterval": integer,
"timedMetadataPid": "string",
"timedMetadataBehavior": enum
]
},
"audioRenditionSets": "string"
},
"audioOnlyHlsSettings": {
"audioTrackType": enum,
"audioGroupId": "string",
"audioOnlyImage": {
"passwordParam": "string",
"uri": "string",
"username": "string"
}
}
},
"nameModifier": "string"
},
"audioDescriptionNames": [
"string"
]
],
"outputGroupSettings": {
"archiveGroupSettings": {
"destination": {
"destinationRefId": "string"
},
"rolloverInterval": integer
},
"mediaPackageGroupSettings": {
"destination": {
"destinationRefId": "string"
}
},
"rtmpGroupSettings": {
"inputLossAction": enum,
"captionData": enum,
"authenticationScheme": enum,
"cacheLength": integer,
"restartDelay": integer,
"cacheFullBehavior": enum
},
"udpGroupSettings": {
"inputLossAction": enum,
"timedMetadataId3Frame": enum,
"timedMetadataId3Period": integer
},
"msSmoothGroupSettings": {
"fragmentLength": integer,
"eventId": "string",
"timestampOffset": "string",
"segmentationMode": enum,
"numRetries": integer,
"eventStopBehavior": enum,
"acquisitionPointId": "string",
"sparseTrackType": enum,
"timestampOffsetMode": enum,
"destination": {
"destinationRefId": "string"
}
},
"147"}
"audioOnlyTimecodeControl": enum,
"connectionRetryInterval": integer,
"filecacheDuration": integer,
"certificateMode": enum,
"inputLossAction": enum,
"sendDelayMs": integer,
"eventIdMode": enum,
"restartDelay": integer,
"streamManifestBehavior": enum,
"hlsGroupSettings": {
  "segmentsPerSubdirectory": integer,
  "ivInManifest": enum,
  "outputSelection": enum,
  "destination": {
    "destinationRefId": "string"
  },
  "encryptionType": enum,
  "indexNumSegments": integer,
  "constantIv": "string",
  "timedMetadata1d3Frame": enum,
  "baseUrlManifest": "string",
  "captionLanguageSetting": enum,
  "minSegmentLength": integer,
  "mode": enum,
  "keyProviderSettings": {
    "staticKeySettings": {
      "staticKeyValue": "string",
      "keyProviderServer": {
        "passwordParam": "string",
        "uri": "string",
        "username": "string"
      }
    }
  },
  "manifestCompression": enum,
  "ivSource": enum,
  "esFileMode": enum,
  "manifestDurationFormat": enum,
  "keyFormatVersions": "string",
  "streamInfResolution": enum,
  "timestampDeltaMilliseconds": integer,
  "baseUrlContent": "string",
  "segmentationMode": enum,
  "captionLanguageMappings": [
    {
      "languageDescription": "string",
      "captionChannel": integer,
      "languageCode": "string"
    }
  ],
  "clientCache": enum,
  "codecSpecification": enum,
  "keepSegments": integer,
  "redundantManifest": enum,
  "timedMetadata1d3Period": integer,
  "programDateTime": enum,
  "directoryStructure": enum,
  "keyFormat": "string",
  "inputLossAction": enum,
  "adMarkers": [enum
  ],
  "programDateTimePeriod": integer,
  "segmentLength": integer,
  "hlsCdnSettings": {
"hlsAkamaiSettings": {
  "salt": "string",
  "httpTransferMode": enum,
  "numRetries": integer,
  "restartDelay": integer,
  "connectionRetryInterval": integer,
  "filecacheDuration": integer,
  "token": "string"
},
"hlsWebdavSettings": {
  "httpTransferMode": enum,
  "numRetries": integer,
  "restartDelay": integer,
  "connectionRetryInterval": integer,
  "filecacheDuration": integer
},
"hlsBasicPutSettings": {
  "numRetries": integer,
  "restartDelay": integer,
  "connectionRetryInterval": integer,
  "filecacheDuration": integer
},
"hlsMediaStoreSettings": {
  "mediaStoreStorageClass": enum,
  "numRetries": integer,
  "restartDelay": integer,
  "connectionRetryInterval": integer,
  "filecacheDuration": integer
},
"iFrameOnlyPlaylists": enum
"frameCaptureGroupSettings": {
  "destination": {
    "destinationRefId": "string"
  }
},
"audioDescriptions": [
  {
    "audioTypeControl": enum,
    "languageCodeControl": enum,
    "remixSettings": {
      "channelMappings": [
        {
          "outputChannel": integer,
          "inputChannelLevels": [
            {
              "inputChannel": integer,
              "gain": integer
            }
          ],
          "channelsOut": integer,
          "channelsIn": integer
        }
      ],
      "audioType": enum,
      "name": "string",
      "codecSettings": {
        "aacSettings": {
          "vbrQuality": enum,
          "codingMode": enum,
          "tbrQuality": enum,
          "codingMode": enum,
          "tbrMode": enum,
          "dynamicRange": enum
        }
      }
    }
  }
]
"profile": enum,
"inputType": enum,
"bitrate": number,
"rawFormat": enum,
"rateControlMode": enum,
"sampleRate": number,
"spec": enum
},
"ac3Settings": {
"drcProfile": enum,
"dialnorm": integer,
"codingMode": enum,
"metadataControl": enum,
"bitrate": number,
"lfeFilter": enum,
"bitstreamMode": enum
},
"eac3Settings": {
"dialnorm": integer,
"passthroughControl": enum,
"drcLine": enum,
"metadataControl": enum,
"bitrate": number,
"ltRtSurroundMixLevel": number,
"surroundExMode": enum,
"lfeControl": enum,
"codingMode": enum,
"surroundMode": enum,
"attenuationControl": enum,
"lfeFilter": enum,
"dcFilter": enum,
"ltRtCenterMixLevel": number,
"phaseControl": enum,
"bitstreamMode": enum,
"stereoDownmix": enum,
"loRoSurroundMixLevel": number,
"drcRf": enum,
"loRoCenterMixLevel": number
},
"passThroughSettings": {
},
"mp2Settings": {
"codingMode": enum,
"bitrate": number,
"sampleRate": number
},
"languageCode": "string",
"streamName": "string",
"audioNormalizationSettings": {
"targetLkfs": number,
"algorithmControl": enum,
"algorithm": enum
},
"audioSelectorName": "string"
},
"availConfiguration": {
"availSettings": {
"scte35TimeSignalApos": {
"adAvailOffset": integer,
"webDeliveryAllowedFlag": enum,
"noRegionalBlackoutFlag": enum
},
"scte35SpliceInsert": {
"adAvailOffset": integer,
"webDeliveryAllowedFlag": enum,
"noRegionalBlackoutFlag": enum
}

captionDescriptions": [
{
"captionSelectorName": "string",
"languageDescription": "string",
"name": "string",
"languageCode": "string",
"destinationSettings": {
"burnInDestinationSettings": {
"xPosition": integer,
"yPosition": integer,
"teletextGridControl": enum,
"backgroundColor": enum,
"backgroundOpacity": integer,
"fontOpacity": integer,
"fontResolution": integer,
"shadowOpacity": integer,
"shadowYOffset": integer,
"outlineSize": integer,
"outlineColor": enum,
"fontSize": "string",
"alignment": enum,
"shadowXOffset": integer,
"shadowColor": enum,
"fontColor": enum,
"font": {
"passwordParam": "string",
"uri": "string",
"username": "string"
}
},
"scte27DestinationSettings": {
},
"teletextDestinationSettings": {
},
"ttmlDestinationSettings": {
"styleControl": enum
},
"smpteTtDestinationSettings": {
},
"webvttDestinationSettings": {
},
"embeddedPlusScte20DestinationSettings": {
},
"dvbSubDestinationSettings": {
"xPosition": integer,
"yPosition": integer,
"teletextGridControl": enum,
"backgroundColor": enum,
"backgroundOpacity": integer,
"fontOpacity": integer,
"fontResolution": integer,
"shadowOpacity": integer,
"shadowYOffset": integer,
"outlineSize": integer,
"outlineColor": enum,
"fontSize": "string",
"alignment": enum,
"shadowXOffset": integer,
"shadowColor": enum,
"fontColor": enum,
"font": {

```
"passwordParam": "string",
"uri": "string",
"username": "string"
},
"embeddedDestinationSettings": {
},
"rtmpCaptionInfoDestinationSettings": {
},
"aribDestinationSettings": {
},
"scte20PlusEmbeddedDestinationSettings": {
}
]
}
"globalConfiguration": {
"inputLossBehavior": {
"inputLossImageType": enum,
"inputLossImageColor": "string",
"inputLossImageSlate": {
"passwordParam": "string",
"uri": "string",
"username": "string"
},
"repeatFrameMsec": integer,
"blackFrameMsec": integer
},
"supportLowFramerateInputs": enum,
"outputLockingMode": enum,
"initialAudioGain": integer,
"inputEndAction": enum,
"outputTimingSource": enum
},
"videoDescriptions": [
{
"respondToAfD": enum,
"scalingBehavior": enum,
"name": "string",
"width": integer,
"sharpness": integer,
"codecSettings": {
"h264Settings": {
"minIInterval": integer,
"slices": integer,
"parNumerator": integer,
"gopSizeUnits": enum,
"subgopLength": enum,
"maxBitrate": integer,
"bitrate": integer,
"bufFillPct": integer,
"temporalAq": enum,
"afdSignaling": enum,
"timecodeInsertion": enum,
"bufSize": integer,
"softness": integer,
"framerateControl": enum,
"gqbrQualityLevel": integer,
"fixedAfD": enum,
"level": enum,
"lookAheadRateControl": enum,
"profile": enum,
"framerateNumerator": integer,
"gopClosedCadence": integer,
"entropyEncoding": enum,
"framerateDenominator": integer,
"spatialAq": enum,
"adaptiveQuantization": enum,
"colorMetadata": enum,
"gopSize": number,
"numRefFrames": integer,
"gopBReference": enum,
"parControl": enum,
"parDenominator": integer,
"syntax": enum,
"sceneChangeDetect": enum,
"scanType": enum,
"flickerAq": enum,
"gopNumBFrames": integer,
"rateControlMode": enum
},
"frameCaptureSettings": {
  "captureInterval": integer
},
"height": integer
},
"availBlanking": {
  "state": enum,
  "availBlankingImage": {
    "passwordParam": "string",
    "uri": "string",
    "username": "string"
  }
},
"blackoutSlate": {
  "networkEndBlackoutImage": {
    "passwordParam": "string",
    "uri": "string",
    "username": "string"
  },
  "networkEndBlackout": enum,
  "networkId": "string",
  "state": enum,
  "blackoutSlateImage": {
    "passwordParam": "string",
    "uri": "string",
    "username": "string"
  }
}
},
"inputSpecification": {
  "codec": enum,
  "resolution": enum,
  "maximumBitrate": enum
}
}

Response Bodies

Example Channel

```json
{
"inputAttachments": [
{
  "inputId": "string",
  "inputAttachmentName": "string",
  "inputSettings": {
    "sourceEndBehavior": enum,
```
"deblockFilter": enum,
"audioSelectors": [
{
  "name": "string",
  "selectorSettings": {
    "audioLanguageSelection": {
      "languageSelectionPolicy": enum,
      "languageCode": "string"
    },
    "audioPidSelection": {
      "pid": integer
    }
  }
},
"networkInputSettings": {
  "hlsInputSettings": {
    "retries": integer,
    "bandwidth": integer,
    "retryInterval": integer,
    "bufferSegments": integer
  },
  "serverValidation": enum
},
"inputFilter": enum,
"videoSelector": {
  "colorSpace": enum,
  "selectorSettings": {
    "videoSelectorPid": {
      "pid": integer
    },
    "videoSelectorProgramId": {
      "programId": integer
    }
  },
  "colorSpaceUsage": enum
},
"filterStrength": integer,
"denoiseFilter": enum,
"captionSelectors": [
{
  "name": "string",
  "languageCode": "string",
  "selectorSettings": {
    "embeddedSourceSettings": {
      "scte20Detection": enum,
      "source608ChannelNumber": integer,
      "convert608To708": enum,
      "source608TrackNumber": integer
    },
    "scte20SourceSettings": {
      "source608ChannelNumber": integer,
      "convert608To708": enum
    },
    "dvbSubSourceSettings": {
      "pid": integer
    },
    "teletextSourceSettings": {
      "pageNumber": "string"
    },
    "aribSourceSettings": {
    },
    "scte27SourceSettings": {
      "pid": integer
    }
  }
}
"destinations": [
  {
    "mediaPackageSettings": [
      {
        "channelId": "string"
      }
    ],
    "settings": [
      {
        "passwordParam": "string",
        "streamName": "string",
        "url": "string",
        "username": "string"
      }
    ],
    "id": "string"
  }
],
"encoderSettings": {
  "timecodeConfig": {
    "syncThreshold": integer,
    "source": enum
  },
  "outputGroups": [
    {
      "outputs": [
        {
          "videoDescriptionName": "string",
          "outputName": "string",
          "captionDescriptionNames": [
            "string"
          ],
          "outputSettings": {
            "rtmpOutputSettings": {
              "certificateMode": enum,
              "numRetries": integer,
              "destination": {
                "destinationRefId": "string"
              },
              "connectionRetryInterval": integer
            },
            "archiveOutputSettings": {
              "extension": "string",
              "containerSettings": {
                "m2tsSettings": {
                  "audioStreamType": enum,
                  "ecmPid": "string",
                  "dvbTeletextPid": "string",
                  "aribCaptionsPidControl": enum,
                  "bitrate": integer,
                  "rateMode": enum,
                  "segmentationTime": number,
                  "audioPids": "string",
                  "audioFramesPerPes": integer,
                  "fragmentTime": number,
                  "ebpLookaheadMs": integer,
                  "ebpAudioInterval": enum,
                  "scte35Pid": "string",
                  "programNum": integer,
                  "pmtInterval": integer,
                  "pcrPeriod": integer,
                },
"segmentationStyle": enum,
"ebif": enum,
"audioBufferModel": enum,
"dvbNitSettings": {
  "networkName": "string",
  "networkId": integer,
  "repInterval": integer
},
"absentInputAudioBehavior": enum,
"timedMetadataBehavior": enum,
"timedMetadataPid": "string",
"pmtPid": "string",
"etvSignalPid": "string",
"bufferModel": enum,
"scte35Control": enum,
"ebpPlacement": enum,
"arib": enum,
"nullPacketBitrate": number,
"dvbSdtSettings": {
  "serviceName": "string",
  "serviceProviderName": "string",
  "repInterval": integer,
  "outputSdt": enum
},
"pcrPid": "string",
"transportStreamId": integer,
"pcrControl": enum,
"videoPid": "string",
"esRateInPes": enum,
"segmentationMarkers": enum,
"dvbTdtSettings": {
  "repInterval": integer
},
"klv": enum,
"ccDescriptor": enum,
"patInterval": integer,
"etvPlatformPid": "string",
"dvbSubPids": "string",
"aribCaptionsPid": "string",
"scte27Pids": "string",
"klvDataPids": "string"
},
"nameModifier": "string"
},
"frameCaptureOutputSettings": {
  "nameModifier": "string"
},
"msSmoothOutputSettings": {
  "nameModifier": "string"
},
"mediaPackageOutputSettings": {
},
"udpOutputSettings": {
  "destination": {
    "destinationRefId": "string"
  },
  "bufferMsec": integer,
  "containerSettings": {
    "m2tsSettings": {
      "audioStreamType": enum,
      "ecmPid": "string",
      "dvbTeletextPid": "string",
      "aribCaptionsPidControl": enum,
      "bitrate": integer,
      "rateMode": enum,
"segmentationTime": number,
"audioPids": "string",
"audioFramesPerPes": integer,
"fragmentTime": number,
"ebpLookaheadMs": integer,
"ebpAudioInterval": enum,
"scte35Pid": "string",
"programNum": integer,
"pmtInterval": integer,
"pcrPeriod": integer,
"segmentationStyle": enum,
"ebp": enum,
"audioBufferModel": enum,
"dvbNitSettings": { 
  "networkName": "string",
  "networkId": integer,
  "repInterval": integer
},
"absentInputAudioBehavior": enum,
"timedMetadataBehavior": enum,
"timedMetadataPid": "string",
"pmtPid": "string",
"etvSignalPid": "string",
"bufferModel": enum,
"scte35Control": enum,
"ebpPlacement": enum,
"arib": enum,
"nullPacketBitrate": number,
"dvbSdtSettings": { 
  "serviceName": "string",
  "serviceProviderName": "string",
  "repInterval": integer,
  "outputSdt": enum
},
"pcrPid": "string",
"transportStreamId": integer,
"pcrControl": enum,
"videoPid": "string",
"esRateInPes": enum,
"segmentationMarkers": enum,
"dvbTdtSettings": { 
  "repInterval": integer
},
"klv": enum,
"ccDescriptor": enum,
"patInterval": integer,
"etvPlatformPid": "string",
"dvbSubPids": "string",
"aribCaptionsPid": "string",
"scte27Pids": "string",
"klvDataPids": "string"
},
"fecOutputSettings": { 
  "rowLength": integer,
  "columnDepth": integer,
  "includeFec": enum
},
"hlsOutputSettings": { 
  "segmentModifier": "string",
  "hlsSettings": { 
    "standardHlsSettings": { 
      "m3u8Settings": { 
        "pmtPid": "string",
        "ecmPid": "string",
        "tsc": integer
      }
    },
    "scte27Settings": { 
      "scte27Pids": "string",
      "ebi": "string",
      "pcrPid": "string",
      "pmtPid": "string",
      "etvPlatformPid": "string",
      "nullPacketBitrate": number,
      "bufferModel": enum,
      "scte35Control": enum,
      "ebpPlacement": enum,
      "arib": enum,
      "nullPacketBitrate": number,
      "dvbSdtSettings": { 
        "serviceName": "string",
        "serviceProviderName": "string",
        "repInterval": integer,
        "outputSdt": enum
      },
      "nullPacketBitrate": number,
      "dvbSdtSettings": { 
        "serviceName": "string",
        "serviceProviderName": "string",
        "repInterval": integer,
        "outputSdt": enum
      }
    }
  }
}
"scte35Behavior": enum,
"pcrPid": "string",
"audioPids": "string",
"audioFramesPerPes": integer,
"scte35Pid": "string",
"transportStreamId": integer,
"pcrControl": enum,
"videoPid": "string",
"pcrPeriod": integer,
"pmtInterval": integer,
"programNum": integer,
"patInterval": integer,
"timedMetadataPid": "string",
"timedMetadataBehavior": enum
},
"audioRenditionSets": "string"
},
"audioOnlyHlsSettings": {
  "audioTrackType": enum,
  "audioGroupId": "string",
  "audioOnlyImage": {
    "passwordParam": "string",
    "uri": "string",
    "username": "string"
  }
},
"nameModifier": "string"
],
"audioDescriptionNames": [
  "string"
]
],
"outputGroupSettings": {
  "archiveGroupSettings": {
    "destination": {
      "destinationRefId": "string"
    },
    "rolloverInterval": integer
  },
  "mediaPackageGroupSettings": {
    "destination": {
      "destinationRefId": "string"
    }
  },
  "rtmpGroupSettings": {
    "inputLossAction": enum,
    "captionData": enum,
    "authenticationScheme": enum,
    "cacheLength": integer,
    "restartDelay": integer,
    "cacheFullBehavior": enum
  },
  "udpGroupSettings": {
    "inputLossAction": enum,
    "timedMetadataId3Frame": enum,
    "timedMetadataId3Period": integer
  },
  "msSmoothGroupSettings": {
    "fragmentLength": integer,
    "eventId": "string",
    "timestampOffset": "string",
    "segmentationMode": enum,
    "numRetries": integer,
"eventStopBehavior": enum,
"acquisitionPointId": "string",
"sparseTrackType": enum,
"timestampOffsetMode": enum,
"destination": {
  "destinationRefId": "string"
},
"audioOnlyTimecodeControl": enum,
"connectionRetryInterval": integer,
"filecacheDuration": integer,
"certificateMode": enum,
"inputLossAction": enum,
"sendDelayMs": integer,
"eventIdMode": enum,
"restartDelay": integer,
"streamManifestBehavior": enum
},
"hlsGroupSettings": {
  "segmentsPerSubdirectory": integer,
  "ivInManifest": enum,
  "outputSelection": enum,
  "destination": {
    "destinationRefId": "string"
  },
  "encryptionType": enum,
  "indexNSegments": integer,
  "constantIv": "string",
  "timedMetadataId3Frame": enum,
  "baseUrlManifest": "string",
  "captionLanguageSetting": enum,
  "minSegmentLength": integer,
  "mode": enum,
  "keyProviderSettings": {
    "staticKeySettings": {
      "staticKeyValue": "string",
      "keyProviderServer": {
        "passwordParam": "string",
        "uri": "string",
        "username": "string"
      }
    }
  },
  "manifestCompression": enum,
  "ivSource": enum,
  "tsFileMode": enum,
  "manifestDurationFormat": enum,
  "streamInfResolution": enum,
  "timestampDeltaMilliseconds": integer,
  "baseUrlContent": "string",
  "segmentationMode": enum,
  "captionLanguageMappings": [
    {
      "languageDescription": "string",
      "captionChannel": integer,
      "languageCode": "string"
    }
  ],
  "clientCache": enum,
  "codecSpecification": enum,
  "keepSegments": integer,
  "redundantManifest": enum,
  "timedMetadataId3Period": integer,
  "programDateTime": enum,
  "directoryStructure": enum,
  "keyFormat": "string"
"inputLossAction": enum,
"adMarkers": [
  enum
],
"programDateTimePeriod": integer,
"segmentLength": integer,
"hlsCdnSettings": {
  "hlsAkamaiSettings": {
    "salt": "string",
    "httpTransferMode": enum,
    "numRetries": integer,
    "restartDelay": integer,
    "connectionRetryInterval": integer,
    "filecacheDuration": integer,
    "token": "string"
  },
  "hlsWebdavSettings": {
    "httpTransferMode": enum,
    "numRetries": integer,
    "restartDelay": integer,
    "connectionRetryInterval": integer,
    "filecacheDuration": integer
  },
  "hlsBasicPutSettings": {
    "numRetries": integer,
    "restartDelay": integer,
    "connectionRetryInterval": integer,
    "filecacheDuration": integer
  },
  "hlsMediaStoreSettings": {
    "mediaStoreStorageClass": enum,
    "numRetries": integer,
    "restartDelay": integer,
    "connectionRetryInterval": integer,
    "filecacheDuration": integer
  }
},
"iFrameOnlyPlaylists": enum
],
"frameCaptureGroupSettings": {
  "destination": {
    "destinationRefId": "string"
  }
},
"name": "string"
],
"audioDescriptions": [
  {
    "audioTypeControl": enum,
    "languageCodeControl": enum,
    "remixSettings": {
      "channelMappings": [
        {"outputChannel": integer,
        "inputChannelLevels": [
          {"inputChannel": integer,
          "gain": integer
        ]}
      }
    }
  }
],
"channelsOut": integer,
"channelsIn": integer
"audioType": enum,
"name": "string",
"codecSettings": {
  "aacSettings": {
    "vbrQuality": enum,
    "codingMode": enum,
    "profile": enum,
    "inputType": enum,
    "bitrate": number,
    "rawFormat": enum,
    "rateControlMode": enum,
    "sampleRate": number,
    "spec": enum
  },
  "ac3Settings": {
    "drcProfile": enum,
    "dialnorm": integer,
    "codingMode": enum,
    "metadataControl": enum,
    "bitrate": number,
    "lfeFilter": enum,
    "bitstreamMode": enum
  },
  "eac3Settings": {
    "dialnorm": integer,
    "passThroughControl": enum,
    "drcLine": enum,
    "metadataControl": enum,
    "bitrate": number,
    "l1RtSurroundMixLevel": number,
    "surroundExMode": enum,
    "lfeControl": enum,
    "codingMode": enum,
    "surroundMode": enum,
    "attenuationControl": enum,
    "lfeFilter": enum,
    "dcrFilter": enum,
    "l1RtCenterMixLevel": number,
    "phaseControl": enum,
    "bitstreamMode": enum,
    "stereoDownmix": enum,
    "loRoSurroundMixLevel": number,
    "drcRf": enum,
    "loRoCenterMixLevel": number
  },
  "passThroughSettings": {
  },
  "mp2Settings": {
    "codingMode": enum,
    "bitrate": number,
    "sampleRate": number
  }
},
"languageCode": "string",
"streamName": "string",
"audioNormalizationSettings": {
  "targetLkfs": number,
  "algorithmControl": enum,
  "algorithm": enum
},
"audioSelectorName": "string"
}
"scte35TimeSignal": {
  "adAvailabilityOffset": integer,
  "webDeliveryAllowedFlag": enum,
  "noRegionalBlackoutFlag": enum
},
"scte35SpliceInsert": {
  "adAvailabilityOffset": integer,
  "webDeliveryAllowedFlag": enum,
  "noRegionalBlackoutFlag": enum
}
},
"captionDescriptions": [
{
  "captionSelectorName": "string",
  "languageDescription": "string",
  "name": "string",
  "languageCode": "string",
  "destinationSettings": {
    "burnInDestinationSettings": {
      "xPosition": integer,
      "backgroundColor": enum,
      "yPosition": integer,
      "teletextGridControl": enum,
      "backgroundColor": integer,
      "fontOpacity": integer,
      "fontResolution": integer,
      "shadowOpacity": integer,
      "shadowYOffset": integer,
      "outlineSize": integer,
      "outlineColor": enum,
      "fontSize": "string",
      "alignment": enum,
      "shadowXOffset": integer,
      "shadowColor": enum,
      "fontName": {
        "passwordParam": "string",
        "uri": "string",
        "username": "string"
      }
    },
    "scte27DestinationSettings": {
    },
    "teletextDestinationSettings": {
    },
    "ttmlDestinationSettings": {
      "styleControl": enum
    },
    "smpteTtDestinationSettings": {
    },
    "webvttDestinationSettings": {
    },
    "embeddedPlusScte20DestinationSettings": {
    },
    "dvbSubDestinationSettings": {
      "xPosition": integer,
      "backgroundColor": enum,
      "yPosition": integer,
      "teletextGridControl": enum,
      "backgroundColor": integer,
      "fontOpacity": integer,
      "fontResolution": integer,
      "shadowOpacity": integer,
      "shadowYOffset": integer,
      "outlineSize": integer,
      "outlineColor": enum,
      "fontSize": "string",
      "alignment": enum,
      "shadowXOffset": integer,
      "shadowColor": enum,
      "fontName": {
        "passwordParam": "string",
        "uri": "string",
        "username": "string"
      }
    }
  }
}
"outlineColor": enum,
"fontSize": "string",
"alignment": enum,
"shadowXOffset": integer,
"shadowColor": enum,
"fontColor": enum,
"font": {
  "passwordParam": "string",
  "uri": "string",
  "username": "string"
},
"embeddedDestinationSettings": {
},
"rtmpCaptionInfoDestinationSettings": {
},
"aribDestinationSettings": {
},
"scte20PlusEmbeddedDestinationSettings": {
}
},
"globalConfiguration": {
  "inputLossBehavior": {
    "inputLossImageType": enum,
    "inputLossImageColor": "string",
    "inputLossImageSlate": {
      "passwordParam": "string",
      "uri": "string",
      "username": "string"
    },
    "repeatFrameMsec": integer,
    "blackFrameMsec": integer
  },
  "supportLowFramerateInputs": enum,
  "outputLockingMode": enum,
  "initialAudioGain": integer,
  "inputEndAction": enum,
  "outputTimingSource": enum
},
"videoDescriptions": [
  {
    "respondToAfD": enum,
    "scalingBehavior": enum,
    "name": "string",
    "width": integer,
    "sharpness": integer,
    "codecSettings": {
      "h264Settings": {
        "minIInterval": integer,
        "slices": integer,
        "parNumerator": integer,
        "gopSizeUnits": enum,
        "subgopLength": enum,
        "maxBitrate": integer,
        "bitrate": integer,
        "bufFillPct": integer,
        "temporalAq": enum,
        "afdSignaling": enum,
        "timecodeInsertion": enum,
        "bufSize": integer,
        "softness": integer,
        "framerateControl": enum,
        "qvbrQualityLevel": integer,
        "fixedAfD": enum,
"level": enum,
"lookAheadRateControl": enum,
"profile": enum,
"framerateNumerator": integer,
"gopClosedCadence": integer,
"entropyEncoding": enum,
"frameCaptureSettings": { 
  "captureInterval": integer
},
"height": integer,
"availBlanking": { 
  "state": enum,
  "availBlankingImage": { 
    "passwordParam": "string",
    "uri": "string",
    "username": "string"
  }
},
"blackoutSlate": { 
  "networkEndBlackoutImage": { 
    "passwordParam": "string",
    "uri": "string",
    "username": "string"
  },
  "networkEndBlackout": enum,
  "networkId": "string",
  "state": enum,
  "blackoutSlateImage": { 
    "passwordParam": "string",
    "uri": "string",
    "username": "string"
  }
},
"egressEndpoints": [ 
  { "sourceIp": "string"
},
"inputSpecification": { 
  "codec": enum,
  "resolution": enum,
  "maximumBitrate": enum
},
"channelClass": enum,
"tags": { 
}
Schemas

"logLevel": enum,
"roleArn": "string",
"name": "string",
"id": "string",
"state": enum,
"pipelinesRunningCount": integer,
"arn": "string"

Example UpdateChannelResultModel

{
  "channel": {
    "inputAttachments": [
      {
        "inputId": "string",
        "inputAttachmentName": "string",
        "inputSettings": {
          "sourceEndBehavior": enum,
          "deblockFilter": enum,
          "audioSelectors": [
            {
              "name": "string",
              "selectorSettings": {
                "audioLanguageSelection": {
                  "languageSelectionPolicy": enum,
                  "languageCode": "string"
                },
                "audioPidSelection": {
                  "pid": integer
                }
              }
            }
          ],
          "networkInputSettings": {
            "hlsInputSettings": {
              "retries": integer,
              "bandwidth": integer,
              "retryInterval": integer,
              "bufferSegments": integer
            },
            "serverValidation": enum
          },
          "inputFilter": enum,
          "videoSelector": {
            "colorSpace": enum,
            "selectorSettings": {
              "videoSelectorPid": {
                "pid": integer
              },
              "videoSelectorProgramId": {
                "programId": integer
              }
            },
            "colorSpaceUsage": enum
          },
          "filterStrength": integer,
          "denoiseFilter": enum,
          "captionSelectors": [
            {
              "name": "string",
              "languageCode": "string",
              "selectorSettings": {
                "embeddedSourceSettings": {
              
            
```
"scte20Detection": enum,
"source608ChannelNumber": integer,
"convert608To708": enum,
"source608TrackNumber": integer
},
"scte20SourceSettings": {
"source608ChannelNumber": integer,
"convert608To708": enum
},
"dvbSubSourceSettings": {
"pid": integer
},
"teletextSourceSettings": {
"pageNumber": "string"
},
"aribSourceSettings": {
},
"scte27SourceSettings": {
"pid": integer
}
}
],
"destinations": [
{
"mediaPackageSettings": [
{
"channelId": "string"
}
],
"settings": [
{
"passwordParam": "string",
"streamName": "string",
"url": "string",
"username": "string"
}
],
"id": "string"
}
],
"encoderSettings": {
"timecodeConfig": {
"syncThreshold": integer,
"source": enum
},
"outputGroups": [
{
"outputs": [
{
"videoDescriptionName": "string",
"outputName": "string",
"captionDescriptionNames": [
"string"
]
},
"outputSettings": {
"rtmpOutputSettings": {
"certificateMode": enum,
"numRetries": integer,
"destination": {
"destinationRefId": "string"
}
},
"connectionRetryInterval": integer
}
"archiveOutputSettings": {
  "extension": "string",
  "containerSettings": {
    "m2tsSettings": {
      "audioStreamType": enum,
      "ecmPid": "string",
      "dvbTeletextPid": "string",
      "aribCaptionsPidControl": enum,
      "bitrate": integer,
      "rateMode": enum,
      "segmentationTime": number,
      "audioPids": "string",
      "audioFramesPerPes": "string",
      "fragmentTime": number,
      "ebpLookaheadMs": integer,
      "ebpAudioInterval": enum,
      "scte35Pid": "string",
      "programNum": integer,
      "pmtInterval": integer,
      "pcrPeriod": integer,
      "segmentationStyle": enum,
      "ebif": enum,
      "audioBufferModel": enum,
      "dvbNitSettings": {
        "networkName": "string",
        "networkId": integer,
        "repInterval": integer
      },
      "absentInputAudioBehavior": enum,
      "timedMetadataBehavior": enum,
      "timedMetadataPid": "string",
      "pmtPid": "string",
      "etvSignalPid": "string",
      "bufferModel": enum,
      "scte35Control": enum,
      "ebpPlacement": enum,
      "arib": enum,
      "nullPacketBitrate": number,
      "dvbSdtSettings": {
        "serviceName": "string",
        "serviceProviderName": "string",
        "repInterval": integer,
        "outputSdt": enum
      },
      "pcrPid": "string",
      "transportStreamId": integer,
      "pcrControl": enum,
      "videoPid": "string",
      "esRateInPes": enum,
      "segmentationMarkers": enum,
      "dvbTdtSettings": {
        "repInterval": integer
      },
      "klv": enum,
      "ccDescriptor": enum,
      "patInterval": integer,
      "etvPlatformPid": "string",
      "dvbSubPids": "string",
      "aribCaptionsPid": "string",
      "scte27Pids": "string",
      "klvDataPids": "string"
    }
  }
},
"nameModifier": "string"}
"frameCaptureOutputSettings": {
    "nameModifier": "string"
},
"msSmoothOutputSettings": {
    "nameModifier": "string"
},
"mediaPackageOutputSettings": {
},
"udpOutputSettings": {
    "destination": {
        "destinationRefId": "string"
    },
    "bufferMsec": integer,
    "containerSettings": {
        "m2tsSettings": {
            "audioStreamType": enum,
            "ecmPid": "string",
            "dvbTeletextPid": "string",
            "aribCaptionsPidControl": enum,
            "bitrate": integer,
            "rateMode": "string",
            "segmentationTime": number,
            "audioPids": "string",
            "audioFramesPerPes": integer,
            "fragmentTime": number,
            "ebpLookaheadMs": integer,
            "ebpAudioInterval": enum,
            "scte35Pid": "string",
            "programNum": integer,
            "pmtInterval": integer,
            "pcrInterval": integer,
            "segmentationStyle": enum,
            "ebif": enum,
            "audioBufferModel": enum,
            "dvbNitSettings": {
                "networkName": "string",
                "networkId": integer,
                "repInterval": integer
            },
            "absentInputAudioBehavior": enum,
            "timedMetadataBehavior": enum,
            "pmtPid": "string",
            "etvSignalPid": "string",
            "bufferModel": enum,
            "scte35Control": enum,
            "ebpPlacement": enum,
            "arib": enum,
            "nullPacketBitrate": number,
            "dvbSdtSettings": {
                "serviceName": "string",
                "serviceProviderName": "string",
                "repInterval": integer,
                "outputsdt": enum
            },
            "pcrPid": "string",
            "transportStreamId": integer,
            "pcrControl": enum,
            "videoPid": "string",
            "esRateInPes": enum,
            "segmentationMarkers": enum,
            "dvbTdtSettings": {
                "repInterval": integer
            },
            "klv": enum,
            "ccDescriptor": enum,
        }
    }
}
"patInterval": integer,
"etvPlatformPid": "string",
"dvbSubPids": "string",
"aribCaptionsPid": "string",
"scte27Pids": "string",
"klvDataPids": "string"}
},
"fecOutputSettings": {
  "rowLength": integer,
  "columnDepth": integer,
  "includeFec": enum
},
"hlsOutputSettings": {
  "segmentModifier": "string",
  "hlsSettings": {
    "standardHlsSettings": {
      "m3u8Settings": {
        "pmtPid": "string",
        "ecmPid": "string",
        "scte35Behavior": enum,
        "pcrPid": "string",
        "audioPids": "string",
        "audioFramesPerPes": integer,
        "scte35Pid": "string",
        "transportStreamId": integer,
        "pcrControl": enum,
        "videoPid": "string",
        "pcrPeriod": integer,
        "pmtInterval": integer,
        "programNum": integer,
        "pcrPid": integer,
        "timedMetadataPid": "string",
        "timedMetadataBehavior": enum
      },
      "audioRenditionSets": "string"
    },
    "audioOnlyHlsSettings": {
      "audioTrackType": enum,
      "audioGroupId": "string",
      "audioOnlyImage": {
        "passwordParam": "string",
        "uri": "string",
        "username": "string"
      }
    }
  }
},
"nameModifier": "string"
},
"audioDescriptionNames": [
  "string"
]
},
"outputGroupSettings": {
  "archiveGroupSettings": {
    "destination": {
      "destinationRefId": "string"
    },
    "rolloverInterval": integer
  },
  "mediaPackageGroupSettings": {
    "destination": {
      "destinationRefId": "string"
    }
  }
}
"tmpGroupSettings": {
  "inputLossAction": enum,
  "captionData": enum,
  "authenticationScheme": enum,
  "cacheLength": integer,
  "restartDelay": integer,
  "cacheFullBehavior": enum
},
"udpGroupSettings": {
  "inputLossAction": enum,
  "timedMetadataId3Frame": enum,
  "timedMetadataId3Period": integer
},
"msSmoothGroupSettings": {
  "fragmentLength": integer,
  "eventId": "string",
  "timestampOffset": "string",
  "segmentationMode": enum,
  "numRetries": integer,
  "eventStopBehavior": enum,
  "acquisitionPointId": "string",
  "sparseTrackType": enum,
  "timestampOffsetMode": enum,
  "destination": {
    "destinationRefId": "string"
  },
  "audioOnlyTimecodeControl": enum,
  "connectionRetryInterval": integer,
  "filecacheDuration": integer,
  "certificateMode": enum,
  "inputLossAction": enum,
  "sendDelayMs": integer,
  "eventIdMode": enum,
  "restartDelay": integer,
  "streamManifestBehavior": enum
},
"hlsGroupSettings": {
  "segmentsPerSubdirectory": integer,
  "ivInManifest": enum,
  "outputSelection": enum,
  "destination": {
    "destinationRefId": "string"
  },
  "encryptionType": enum,
  "indexNSegments": integer,
  "constantIv": "string",
  "timedMetadataId3Frame": enum,
  "baseUrlManifest": "string",
  "captionLanguageSetting": enum,
  "minSegmentLength": integer,
  "mode": enum,
  "keyProviderSettings": {
    "staticKeySettings": {
      "staticKeyValue": "string",
      "keyProviderServer": {
        "passwordParam": "string",
        "uri": "string",
        "username": "string"
      }
    }
  },
  "manifestCompression": enum,
  "ivSource": enum,
  "tsFileMode": enum,
"manifestDurationFormat": enum,
"keyFormatVersions": "string",
"streamInfResolution": enum,
"timestampDeltaMilliseconds": integer,
"baseUrlContent": "string",
"segmentationMode": enum,
"captionLanguageMappings": [
  {
    "languageDescription": "string",
    "captionChannel": integer,
    "languageCode": "string"
  }
],
"clientCache": enum,
"codecSpecification": enum,
"keepSegments": integer,
"redundantManifest": enum,
"timedMetadataId3Period": integer,
"programDateTime": enum,
"directoryStructure": enum,
"keyFormat": "string",
"inputLossAction": enum,
"adMarkers": [ enum
],
"programDateTimePeriod": integer,
"segmentLength": integer,
"hlsCdnSettings": { hlsAkamaiSettings": {
  "salt": "string",
  "httpTransferMode": enum,
  "numRetries": integer,
  "restartDelay": integer,
  "connectionRetryInterval": integer,
  "filecacheDuration": integer,
  "token": "string"
},
"hlsWebdavSettings": { "httpTransferMode": enum,
  "numRetries": integer,
  "restartDelay": integer,
  "connectionRetryInterval": integer,
  "filecacheDuration": integer }
},
"hlsBasicPutSettings": {
  "numRetries": integer,
  "restartDelay": integer,
  "connectionRetryInterval": integer,
  "filecacheDuration": integer }
},
"hlsMediaStoreSettings": {
  "mediaStoreStorageClass": enum,
  "numRetries": integer,
  "restartDelay": integer,
  "connectionRetryInterval": integer,
  "filecacheDuration": integer }
},
"iFrameOnlyPlaylists": enum
},
"frameCaptureGroupSettings": {
  "destination": {
    "destinationRefId": "string"
  }
}]
}
"name": "string"
}
]

"audioDescriptions": [
{
  "audioTypeControl": enum,
  "languageCodeControl": enum,
  "remixSettings": {
    "channelMappings": [  
      {  
        "outputChannel": integer,
        "inputChannelLevels": [
          {  
            "inputChannel": integer,
            "gain": integer
          }
        ]
      }
    ],
    "channelsOut": integer,
    "channelsIn": integer
  },
  "audioType": enum,
  "name": "string",
  "codecSettings": {
    "aacSettings": {
      "vbrQuality": enum,
      "codingMode": enum,
      "profile": enum,
      "inputType": enum,
      "bitrate": number,
      "rawFormat": enum,
      "rateControlMode": enum,
      "sampleRate": number,
      "spec": enum
    },
    "ac3Settings": {
      "drcProfile": enum,
      "dialnorm": integer,
      "codingMode": enum,
      "metadataControl": enum,
      "bitrate": number,
      "lfeFilter": enum,
      "bitstreamMode": enum
    },
    "eac3Settings": {
      "dialnorm": integer,
      "passsthroughControl": enum,
      "drcLine": enum,
      "metadataControl": enum,
      "bitrate": number,
      "ltRtSurroundMixLevel": number,
      "surroundExMode": enum,
      "lfeControl": enum,
      "codingMode": enum,
      "surroundMode": enum,
      "attenuationControl": enum,
      "lfeFilter": enum,
      "dcFilter": enum,
      "ltRtCenterMixLevel": number,
      "phaseControl": enum,
      "bitstreamMode": enum,
      "stereoDownmix": enum,
      "loRoSurroundMixLevel": number,
      "drcRf": enum,
      "loRoCenterMixLevel": number
    }
  }
]
"passThroughSettings": {}
"mp2Settings": {
  "codingMode": enum,
  "bitrate": number,
  "sampleRate": number
},
"languageCode": "string",
"streamName": "string",
"audioNormalizationSettings": {
  "targetLkfs": number,
  "algorithmControl": enum,
  "algorithm": enum
},
"audioSelectorName": "string"
},
"availConfiguration": {
  "availSettings": {
    "scte35TimeSignalApos": {
      "adAvailOffset": integer,
      "webDeliveryAllowedFlag": enum,
      "noRegionalBlackoutFlag": enum
    },
    "scte35SpliceInsert": {
      "adAvailOffset": integer,
      "webDeliveryAllowedFlag": enum,
      "noRegionalBlackoutFlag": enum
    }
  }
},
"captionDescriptions": [
{ "captionSelectorName": "string",
  "languageDescription": "string",
  "name": "string",
  "languageCode": "string",
  "destinationSettings": {
    "burnInDestinationSettings": {
      "xPosition": integer,
      "backgroundColor": enum,
      "yPosition": integer,
      "teletextGridControl": enum,
      "backgroundOpacity": integer,
      "fontOpacity": integer,
      "fontResolution": integer,
      "shadowOpacity": integer,
      "shadowYOffset": integer,
      "outlineSize": integer,
      "outlineColor": enum,
      "fontSize": "string",
      "alignment": enum,
      "shadowXOffset": integer,
      "shadowColor": enum,
      "fontColor": enum,
      "font": {
        "passwordParam": "string",
        "uri": "string",
        "username": "string"
      }
    },
    "scte27DestinationSettings": {
    }
  },
  "teletextDestinationSettings": {
  }
}]

173
"ttmlDestinationSettings": { "styleControl": enum },
"smpteTtDestinationSettings": { },
"webvttDestinationSettings": { },
"embeddedPlusScte20DestinationSettings": { },
"dvbSubDestinationSettings": { "xPosition": integer, "backgroundColor": enum, "yPosition": integer, "teletextGridControl": enum, "backgroundOpacity": integer, "fontOpacity": integer, "fontResolution": integer, "shadowOpacity": integer, "shadowYOffset": integer, "outlineSize": integer, "outlineColor": enum, "fontSize": "string", "alignment": enum, "shadowXOffset": integer, "shadowColor": enum, "fontColor": enum, "font": { "passwordParam": "string", "uri": "string", "username": "string" } },
"embeddedDestinationSettings": { },
"rtmpCaptionInfoDestinationSettings": { },
"ariibDestinationSettings": { },
"scte20PlusEmbeddedDestinationSettings": { }
"globalConfiguration": { "inputLossBehavior": { "inputLossImageType": enum, "inputLossImageColor": "string", "inputLossImageSlate": { "passwordParam": "string", "uri": "string", "username": "string" },
"repeatFrameMsec": integer, "blackFrameMsec": integer },
"supportLowFramerateInputs": enum, "outputLockingMode": enum, "initialAudioGain": integer, "inputEndAction": enum, "outputTimingSource": enum },
"videoDescriptions": [ { "respondToAfd": enum, "scalingBehavior": enum,} ]}
"name": "string",
"width": integer,
"sharpness": integer,
"codecSettings": {
  "h264Settings": {
    "minIInterval": integer,
    "slices": integer,
    "parNumerator": integer,
    "gopSizeUnits": enum,
    "subgopLength": enum,
    "maxBitrate": integer,
    "bitrate": integer,
    "bufFillPct": integer,
    "temporalAq": enum,
    "afdSignaling": enum,
    "timecodeInsertion": enum,
    "bufSize": integer,
    "softness": integer,
    "framerateControl": enum,
    "qvbrQualityLevel": integer,
    "fixedAfd": enum,
    "level": enum,
    "lookAheadRateControl": enum,
    "profile": enum,
    "framerateNumerator": integer,
    "gopClosedCadence": integer,
    "entropyEncoding": enum,
    "framerateDenominator": integer,
    "spatialAq": enum,
    "adaptiveQuantization": enum,
    "colorMetadata": enum,
    "gopSize": number,
    "numRefFrames": integer,
    "gopBReference": enum,
    "parControl": enum,
    "parDenominator": integer,
    "syntax": enum,
    "sceneChangeDetect": enum,
    "scanType": enum,
    "flickerAq": enum,
    "gopNumBFrames": integer,
    "rateControlMode": enum
  },
  "frameCaptureSettings": {
    "captureInterval": integer
  }
},
"height": integer
},
"availBlanking": {
  "state": enum,
  "availBlankingImage": {
    "passwordParam": "string",
    "uri": "string",
    "username": "string"
  }
},
"blackoutSlate": {
  "networkEndBlackoutImage": {
    "passwordParam": "string",
    "uri": "string",
    "username": "string"
  },
  "networkEndBlackout": enum,
  "networkid": "string"
Example InvalidRequest

```json
{
  "message": "string"
}
```

Example AccessDenied

```json
{
  "message": "string"
}
```

Example ResourceNotFound

```json
{
  "message": "string"
}
```

Example ResourceConflict

```json
{
  "message": "string"
}
```

Example ChannelConfigurationValidationError

```json
{
```
"validationErrors": [  
    {  
      "errorMessage": "string",
      "elementPath": "string"
    },
    "message": "string"
  ],

Example LimitExceeded

{
  "message": "string"
}

Example InternalServiceError

{
  "message": "string"
}

Example BadGatewayException

{
  "message": "string"
}

Example GatewayTimeoutException

{
  "message": "string"
}

Properties

AacCodingMode

Aac Coding Mode

    AD_RECEIVER_MIX
    CODING_MODE_1_0
    CODING_MODE_1_1
    CODING_MODE_2_0
    CODING_MODE_5_1

AacInputType

Aac Input Type

    BROADCASTER_MIXED_AD
    NORMAL

177
**AacProfile**

Aac Profile

- HEV1
- HEV2
- LC

**AacRateControlMode**

Aac Rate Control Mode

- CBR
- VBR

**AacRawFormat**

Aac Raw Format

- LATM_LOAS
- NONE

**AacSettings**

Aac Settings

**vbrQuality**

VBR Quality Level - Only used if rateControlMode is VBR.

- Type: AacVbrQuality (p. 179)
- Required: False

**codingMode**

Mono, Stereo, or 5.1 channel layout. Valid values depend on rate control mode and profile. The adReceiverMix 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.

- Type: AacCodingMode (p. 177)
- Required: False

**profile**

AAC Profile.

- Type: AacProfile (p. 178)
- Required: False

**inputType**

Set to "broadcasterMixedAd" when input contains pre-mixed main audio + AD (narration) as a stereo pair. The Audio Type field (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. The values in audioTypeControl and audioType (in AudioDescription) are ignored when set to broadcasterMixedAd. Leave set to "normal" when input does not contain pre-mixed audio + AD.

Type: AacInputType (p. 177)
Required: False

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

Type: number
Required: False

rawFormat
Sets LATM / LOAS AAC output for raw containers.

Type: AacRawFormat (p. 178)
Required: False

rateControlMode
Rate Control Mode.

Type: AacRateControlMode (p. 178)
Required: False

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

Type: number
Required: False

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

Type: AacSpec (p. 179)
Required: False

AacSpec
Aac Spec

MPEG2
MPEG4

AacVbrQuality
Aac Vbr Quality
HIGH
LOW
MEDIUM_HIGH
MEDIUM_LOW

**Ac3BitstreamMode**

Ac3 Bitstream Mode

COMMENTARY
COMPLETE_MAIN
DIALOGUE
EMERGENCY
HEARING_IMPAIRED
MUSIC_AND_EFFECTS
VISUALLY_IMPAIRED
VOICE_OVER

**Ac3CodingMode**

Ac3 Coding Mode

CODING_MODE_1_0
CODING_MODE_1_1
CODING_MODE_2_0
CODING_MODE_3_2_LFE

**Ac3DrcProfile**

Ac3 Drc Profile

FILM_STANDARD
NONE

**Ac3LfeFilter**

Ac3 Lfe Filter

DISABLED
ENABLED

**Ac3MetadataControl**

Ac3 Metadata Control

FOLLOW_INPUT
USE_CONFIGURED

**Ac3Settings**

Ac3 Settings
**drcProfile**

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

*Type: Ac3DrcProfile (p. 180)*

*Required: False*

**dialnorm**

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

*Type: integer*

*Required: False*

*Minimum: 1*

*Maximum: 31*

**codingMode**

Dolby Digital coding mode. Determines number of channels.

*Type: Ac3CodingMode (p. 180)*

*Required: False*

**metadataControl**

When set to "followInput", 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.

*Type: Ac3MetadataControl (p. 180)*

*Required: False*

**bitrate**

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

*Type: number*

*Required: False*

**lfeFilter**

When set to enabled, applies a 120Hz lowpass filter to the LFE channel prior to encoding. Only valid in codingMode32Lfe mode.

*Type: Ac3LfeFilter (p. 180)*

*Required: False*

**bitstreamMode**

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

*Type: Ac3BitstreamMode (p. 180)*
Required: False

AccessDenied

message

Type: string
Required: False

AfdSignaling

Afd Signaling

AUTO
FIXED
NONE

ArchiveContainerSettings

Archive Container Settings

m2tsSettings

Type: M2tsSettings (p. 251)
Required: False

ArchiveGroupSettings

Archive Group Settings

destination

A directory and base filename where archive files should be written.

Type: OutputLocationRef (p. 269)
Required: True

rolloverInterval

Number of seconds to write to archive file before closing and starting a new one.

Type: integer
Required: False
Minimum: 1

ArchiveOutputSettings

Archive Output Settings

extension

Output file extension. If excluded, this will be auto-selected from the container type.
Type: string  
Required: False

containerSettings
Settings specific to the container type of the file.

   Type: ArchiveContainerSettings (p. 182)  
   Required: True

nameModifier
String concatenated to the end of the destination filename. Required for multiple outputs of the same type.

   Type: string  
   Required: False

AribDestinationSettings
Arib Destination Settings

AribSourceSettings
Arib Source Settings

AudioChannelMapping
Audio Channel Mapping

outputChannel
The index of the output channel being produced.

   Type: integer  
   Required: True  
   Minimum: 0  
   Maximum: 7

inputChannelLevels
Indices and gain values for each input channel that should be remixed into this output channel.

   Type: Array of type InputChannelLevel (p. 242)  
   Required: True

AudioCodecSettings
Audio Codec Settings

aacSettings

   Type: AacSettings (p. 178)  
   Required: False
ac3Settings

- **Type:** Ac3Settings (p. 180)
- **Required:** False

eac3Settings

- **Type:** Eac3Settings (p. 210)
- **Required:** False

passThroughSettings

- **Type:** PassThroughSettings (p. 270)
- **Required:** False

mp2Settings

- **Type:** Mp2Settings (p. 262)
- **Required:** False

AudioDescription

Audio Description

**audioTypeControl**

Determines how audio type is determined. followInput: If the input contains an ISO 639 audioType, then that value is passed through to the output. If the input contains no ISO 639 audioType, the value in Audio Type is included in the output. useConfigured: The value in Audio Type is included in the output. Note that this field and audioType are both ignored if inputType is broadcasterMixedAd.

- **Type:** AudioDescriptionAudioTypeControl (p. 185)
- **Required:** False

**languageCodeControl**

Choosing followInput will cause the ISO 639 language code of the output to follow the ISO 639 language code of the input. The languageCode will be used when useConfigured is set, or when followInput is selected but there is no ISO 639 language code specified by the input.

- **Type:** AudioDescriptionLanguageCodeControl (p. 186)
- **Required:** False

**remixSettings**

Settings that control how input audio channels are remixed into the output audio channels.

- **Type:** RemixSettings (p. 270)
- **Required:** False

**audioType**

Applies only if audioTypeControl is useConfigured. The values for audioType are defined in ISO-IEC 13818-1.
Properties

**Type**: AudioType (p. 189)
**Required**: False

**name**
The name of this AudioDescription. Outputs will use this name to uniquely identify this AudioDescription. Description names should be unique within this Live Event.

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

**codecSettings**
Audio codec settings.

**Type**: AudioCodecSettings (p. 183)
**Required**: False

**languageCode**
Indicates the language of the audio output track. Only used if languageControlMode is useConfigured, or there is no ISO 639 language code specified in the input.

**Type**: string
**Required**: False
**MinLength**: 3
**MaxLength**: 3

**streamName**
Used for MS Smooth and Apple HLS outputs. Indicates the name displayed by the player (eg. English, or Director Commentary).

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

**audioNormalizationSettings**
Advanced audio normalization settings.

**Type**: AudioNormalizationSettings (p. 187)
**Required**: False

**audioSelectorName**
The name of the AudioSelector used as the source for this AudioDescription.

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

**AudioDescriptionAudioTypeControl**
Audio Description Audio Type Control
**AudioDescriptionLanguageCodeControl**

Audio Description Language Code Control

```text
FOLLOW_INPUT
USE_CONFIGURED
```

**AudioLanguageSelection**

Audio Language Selection

**languageSelectionPolicy**

When set to "strict", the transport stream demux strictly identifies audio streams by their language descriptor. If a PMT update occurs such that an audio stream matching the initially selected language is no longer present then mute will be encoded until the language returns. If "loose", then on a PMT update the demux will choose another audio stream in the program with the same stream type if it can't find one with the same language.

- **Type**: AudioLanguageSelectionPolicy (p. 186)
- **Required**: False

**languageCode**

Selects a specific three-letter language code from within an audio source.

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

**AudioLanguageSelectionPolicy**

Audio Language Selection Policy

- LOOSE
- STRICT

**AudioNormalizationAlgorithm**

Audio Normalization Algorithm

- ITU_1770_1
- ITU_1770_2

**AudioNormalizationAlgorithmControl**

Audio Normalization Algorithm Control

- CORRECT_AUDIO

---

186
**AudioNormalizationSettings**

Audio Normalization Settings

**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
- **Minimum**: -59.0
- **Maximum**: 0.0

**algorithmControl**

When set to correctAudio the output audio is corrected using the chosen algorithm. If set to measureOnly, the audio will be measured but not adjusted.

- **Type**: AudioNormalizationAlgorithmControl (p. 186)
- **Required**: False

**algorithm**

Audio normalization algorithm to use. itu17701 conforms to the CALM Act specification, itu17702 conforms to the EBU R-128 specification.

- **Type**: AudioNormalizationAlgorithm (p. 186)
- **Required**: False

**AudioOnlyHlsSettings**

Audio Only Hls Settings

**audioTrackType**

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=NO 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

- **Type**: AudioOnlyHlsTrackType (p. 188)
- **Required**: False

**audioGroupId**

Specifies the group to which the audio Rendition belongs.

- **Type**: string
Properties

**Required**: False

**audioOnlyImage**

For use with an audio only Stream. Must be a .jpg or .png file. If given, this image will be used as the cover-art for the audio only output. Ideally, it should be formatted for an iPhone screen for two reasons. The iPhone does not resize the image, it crops a centered image on the top/bottom and left/right. Additionally, this image file gets saved bit-for-bit into every 10-second segment file, so will increase bandwidth by \{image file size\} * \{segment count\} * \{user count\}.

**Type**: InputLocation (p. 243)
**Required**: False

**AudioOnlyHlsTrackType**

Audio Only Hls Track Type

- ALTERNATE_AUDIO_AUTO_SELECT
- ALTERNATE_AUDIO_AUTO_SELECT_DEFAULT
- ALTERNATE_AUDIO_NOT_AUTO_SELECT
- AUDIO_ONLY_VARIANT_STREAM

**AudioPidSelection**

Audio Pid Selection

**pid**

Selects a specific PID from within a source.

**Type**: integer
**Required**: True
**Minimum**: 0
**Maximum**: 8191

**AudioSelector**

Audio Selector

**name**

The name of this AudioSelector. AudioDescriptions will use this name to uniquely identify this Selector. Selector names should be unique per input.

**Type**: string
**Required**: True
**MinLength**: 1

**selectorSettings**

The audio selector settings.

**Type**: AudioSelectorSettings (p. 189)
**Required**: False
**AudioSelectorSettings**

Audio Selector Settings

**audioLanguageSelection**

*Type:* AudioLanguageSelection (p. 186)  
*Required:* False

**audioPidSelection**

*Type:* AudioPidSelection (p. 188)  
*Required:* False

**AudioType**

Audio Type

CLEAN_EFFECTS  
HEARING_IMPAIRED  
UNDEFINED  
VISUAL_IMPAIRED_COMMENTARY

**AuthenticationScheme**

Authentication Scheme

AKAMAI  
COMMON

**AvailBlanking**

Avail Blanking

**state**

When set to enabled, causes video, audio and captions to be blanked when insertion metadata is added.

*Type:* AvailBlankingState (p. 189)  
*Required:* False

**availBlankingImage**

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

*Type:* InputLocation (p. 243)  
*Required:* False

**AvailBlankingState**

Avail Blanking State

DISABLED
ENABLED

AvailConfiguration
Avail Configuration

availSettings
Ad avail settings.
  Type: AvailSettings (p. 190)
  Required: False

AvailSettings
Avail Settings

scte35TimeSignalApos
  Type: Scte35TimeSignalApos (p. 275)
  Required: False

scte35SpliceInsert
  Type: Scte35SpliceInsert (p. 274)
  Required: False

BadGatewayException

message
  Type: string
  Required: False

BlackoutSlate
Blackout Slate

networkEndBlackoutImage
Path to local file to use as Network End Blackout image. Image will be scaled to fill the entire output raster.
  Type: InputLocation (p. 243)
  Required: False

networkEndBlackout
Setting to enabled causes the encoder to blackout the video, audio, and captions, and raise the "Network Blackout Image" slate when an SCTE104/35 Network End Segmentation Descriptor is encountered. The blackout will be lifted when the Network Start Segmentation Descriptor is encountered. The Network End and Network Start descriptors must contain a network ID that matches the value entered in "Network ID".
**Properties**

**Type**: BlackoutSlateNetworkEndBlackout (p. 191)
**Required**: False

**networkId**

Provides Network ID that matches EIDR ID format (e.g., "10.XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-C").

**Type**: string
**Required**: False
**MinLength**: 34
**MaxLength**: 34

**state**

When set to enabled, causes video, audio and captions to be blanked when indicated by program metadata.

**Type**: BlackoutSlateState (p. 191)
**Required**: False

**blackoutSlateImage**

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

**Type**: InputLocation (p. 243)
**Required**: False

**BlackoutSlateNetworkEndBlackout**

Blackout Slate Network End Blackout

- DISABLED
- ENABLED

**BlackoutSlateState**

Blackout Slate State

- DISABLED
- ENABLED

**BurnInAlignment**

Burn In Alignment

- CENTERED
- LEFT
- SMART

**BurnInBackgroundColor**

Burn In Background Color
BLACK
NONE
WHITE

**BurnInDestinationSettings**

Burn In Destination Settings

**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 xPosition is provided, the horizontal caption position will be determined by the alignment parameter. All burn-in and DVB-Sub font settings must match.

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

**backgroundColor**

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

**Type:** BurnInBackgroundColor (p. 191)  
**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 yPosition is provided, the caption will be positioned towards the bottom of the output. All burn-in and DVB-Sub font settings must match.

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

**teletextGridControl**

Controls whether a fixed grid size will be used to generate the output subtitles bitmap. Only applicable for Teletext inputs and DVB-Sub/Burn-in outputs.

**Type:** BurnInTeletextGridControl (p. 195)  
**Required:** False

**backgroundOpacity**

Specifies the opacity of the background rectangle. 255 is opaque; 0 is transparent. Leaving this parameter out 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: False
   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 out 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

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: 0
   Maximum: 10

outlineColor

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.
**Type:** BurnInOutlineColor (p. 195)
**Required:** False

**fontSize**
When set to 'auto' fontSize will scale depending on the size of the output. Giving a positive integer will specify the exact font size in points. All burn-in and DVB-Sub font settings must match.

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

**alignment**
If no explicit xPosition or yPosition 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. Selecting "smart" justification will left-justify live subtitles and center-justify pre-recorded subtitles. All burn-in and DVB-Sub font settings must match.

**Type:** BurnInAlignment (p. 191)
**Required:** False

**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

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

**Type:** BurnInShadowColor (p. 195)
**Required:** False

**fontColor**
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.

**Type:** BurnInFontColor (p. 195)
**Required:** False

**font**
External font file used for caption burn-in. File extension must be 'ttf' or 'tte'. Although the user can select output fonts for many different types of input captions, embedded, STL and teletext sources use a strict grid system. Using external fonts with these caption sources could cause unexpected display of proportional fonts. All burn-in and DVB-Sub font settings must match.
Properties

**Type:** InputLocation (p. 243)
**Required:** False

**BurnInFontColor**

Burn In Font Color
- BLACK
- BLUE
- GREEN
- RED
- WHITE
- YELLOW

**BurnInOutlineColor**

Burn In Outline Color
- BLACK
- BLUE
- GREEN
- RED
- WHITE
- YELLOW

**BurnInShadowColor**

Burn In Shadow Color
- BLACK
- NONE
- WHITE

**BurnInTeletextGridControl**

Burn In Teletext Grid Control
- FIXED
- SCALED

**CaptionDescription**

Output groups for this Live Event. Output groups contain information about where streams should be distributed.

**captionSelectorName**

Specifies which input caption selector to use as a caption source when generating output captions. This field should match a captionSelector name.

**Type:** string
**Required:** True
languageDescription

Human readable information to indicate captions available for players (eg. English, or Spanish).

Type: string
Required: False

name

Name of the caption description. Used to associate a caption description with an output. Names must be unique within an event.

Type: string
Required: True

languageCode


Type: string
Required: False

destinationSettings

Additional settings for captions destination that depend on the destination type.

Type: CaptionDestinationSettings (p. 196)
Required: False

CaptionDestinationSettings

Caption Destination Settings

burnInDestinationSettings

Type: BurnInDestinationSettings (p. 192)
Required: False

scte27DestinationSettings

Type: Scte27DestinationSettings (p. 274)
Required: False

teletextDestinationSettings

Type: TeletextDestinationSettings (p. 278)
Required: False

ttmlDestinationSettings

Type: TtmlDestinationSettings (p. 278)
Required: False
Properties

smpteTtDestinationSettings
- Type: SmpteTtDestinationSettings (p. 277)
- Required: False

webvttDestinationSettings
- Type: WebvttDestinationSettings (p. 285)
- Required: False

embeddedPlusScte20DestinationSettings
- Type: EmbeddedPlusScte20DestinationSettings (p. 213)
- Required: False

dvbSubDestinationSettings
- Type: DvbSubDestinationSettings (p. 204)
- Required: False

embeddedDestinationSettings
- Type: EmbeddedDestinationSettings (p. 213)
- Required: False

rtmpCaptionInfoDestinationSettings
- Type: RtmpCaptionInfoDestinationSettings (p. 271)
- Required: False

aribDestinationSettings
- Type: AribDestinationSettings (p. 183)
- Required: False

scte20PlusEmbeddedDestinationSettings
- Type: Scte20PlusEmbeddedDestinationSettings (p. 273)
- Required: False

CaptionLanguageMapping
Maps a caption channel to an ISO 693-2 language code (http://www.loc.gov/standards/iso639-2), with an optional description.

languageDescription
Textual description of language
- Type: string
- Required: True
- MinLength: 1
captionChannel

The closed caption channel being described by this CaptionLanguageMapping. Each channel mapping must have a unique channel number (maximum of 4)

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

languageCode

Three character ISO 639-2 language code (see http://www.loc.gov/standards/iso639-2)

- **Type:** string
- **Required:** True
- **MinLength:** 3
- **MaxLength:** 3

CaptionSelector

Output groups for this Live Event. Output groups contain information about where streams should be distributed.

name

Name identifier for a caption selector. This name is used to associate this caption selector with one or more caption descriptions. Names must be unique within an event.

- **Type:** string
- **Required:** True
- **MinLength:** 1

languageCode

When specified this field indicates the three letter language code of the caption track to extract from the source.

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

selectorSettings

Caption selector settings.

- **Type:** `CaptionSelectorSettings` (p. 198)
- **Required:** False

CaptionSelectorSettings

Caption Selector Settings

embeddedSourceSettings

- **Type:** `EmbeddedSourceSettings` (p. 213)
Properties

Required: False

scte20SourceSettings

Type: Scte20SourceSettings (p. 273)
Required: False

dvbSubSourceSettings

Type: DvbSubSourceSettings (p. 207)
Required: False

teletextSourceSettings

Type: TeletextSourceSettings (p. 278)
Required: False

aribSourceSettings

Type: AribSourceSettings (p. 183)
Required: False

scte27SourceSettings

Type: Scte27SourceSettings (p. 274)
Required: False

Channel

inputAttachments

List of input attachments for channel.

Type: Array of type InputAttachment (p. 242)
Required: False

destinations

A list of destinations of the channel. For UDP outputs, there is one destination per output. For other types (HLS, for example), there is one destination per packager.

Type: Array of type OutputDestination (p. 267)
Required: False

encoderSettings

Type: EncoderSettings (p. 214)
Required: False

egressEndpoints

The endpoints where outgoing connections initiate from
Properties

**Type**
Array of type `ChannelEgressEndpoint (p. 201)`
**Required**: False

**inputSpecification**

**Type**: `InputSpecification (p. 247)`
**Required**: False

**channelClass**
The class for this channel. STANDARD for a channel with two pipelines or SINGLE_PIPELINE for a channel with one pipeline.

**Type**: `ChannelClass (p. 201)`
**Required**: False

**tags**
A collection of key-value pairs.

**Type**: `Tags (p. 277)`
**Required**: False

**logLevel**
The log level being written to CloudWatch Logs.

**Type**: `LogLevel (p. 248)`
**Required**: False

**roleArn**
The Amazon Resource Name (ARN) of the role assumed when running the Channel.

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

**name**
The name of the channel. (user-mutable)

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

**id**
The unique id of the channel.

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

**state**

**Type**: `ChannelState (p. 201)`
**Required**: False

**pipelinesRunningCount**

The number of currently healthy pipelines.

**Type**: integer

**Required**: False

**arn**

The unique arn of the channel.

**Type**: string

**Required**: False

**ChannelClass**

A standard channel has two encoding pipelines and a single pipeline channel only has one.

- STANDARD
- SINGLE_PIPELINE

**ChannelConfigurationValidationError**

**validationErrors**

A collection of validation error responses.

**Type**: Array of type ValidationError (p. 281)

**Required**: False

**message**

**Type**: string

**Required**: False

**ChannelEgressEndpoint**

**sourceIp**

Public IP of where a channel's output comes from.

**Type**: string

**Required**: False

**ChannelState**

- CREATING
- CREATE_FAILED
- IDLE
Properties

STARTING
RUNNING
RECOVERING
STOPPING
DELETING
DELETED
UPDATING
UPDATE_FAILED

DvbNitSettings

DVB Network Information Table (NIT)

networkName

The network name text placed in the networkNameDescriptor inside the Network Information Table. Maximum length is 256 characters.

Type: string
Required: True
MinLength: 1
MaxLength: 256

networkId

The numeric value placed in the Network Information Table (NIT).

Type: integer
Required: True
Minimum: 0
Maximum: 65536

repInterval

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

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

DvbSdtOutputSdt

Dvb Sdt Output Sdt

SDT_FOLLOW
SDT_FOLLOW_IF_PRESENT
SDT_MANUAL
SDT_NONE

DvbSdtSettings

DVB Service Description Table (SDT)
serviceName

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

Type: string
Required: False
MinLength: 1
MaxLength: 256

serviceProviderName

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

Type: string
Required: False
MinLength: 1
MaxLength: 256

repInterval

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

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

outputSdt

Selects method of inserting SDT information into output stream. The sdtFollow setting copies SDT information from input stream to output stream. The sdtFollowIfPresent setting 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. The sdtManual setting means user will enter the SDT information. The sdtNone setting means output stream will not contain SDT information.

Type: DvbSdtOutputSdt (p. 202)
Required: False

DvbSubDestinationAlignment

Dvb Sub Destination Alignment

CENTERED
LEFT
SMART

DvbSubDestinationBackgroundColor

Dvb Sub Destination Background Color

BLACK
NONE
DvbSubDestinationFontColor

Dvb Sub Destination Font Color

BLACK
BLUE
GREEN
RED
WHITE
YELLOW

DvbSubDestinationOutlineColor

Dvb Sub Destination Outline Color

BLACK
BLUE
GREEN
RED
WHITE
YELLOW

DvbSubDestinationSettings

Dvb Sub Destination Settings

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 xPosition 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

backgroundColor

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

Type: DvbSubDestinationBackgroundColor (p. 203)
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 yPosition 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

**teletextGridControl**

Controls whether a fixed grid size will be used to generate the output subtitles bitmap. Only applicable for Teletext inputs and DVB-Sub/Burn-in outputs.

- **Type**: DvbSubDestinationTeletextGridControl (p. 207)
  - **Required**: False

**backgroundColor**

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**: False
  - **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

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: 0
Maximum: 10

outlineColor

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.

Type: DvbSubDestinationOutlineColor (p. 204)
Required: False

fontSize

When set to auto fontSize will scale depending on the size of the output. Giving a positive integer will specify the exact font size in points. All burn-in and DVB-Sub font settings must match.

Type: string
Required: False

alignment

If no explicit xPosition or yPosition 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. Selecting "smart" justification will left-justify live subtitles and center-justify pre-recorded subtitles. This option is not valid for source captions that are STL or 608/embedded. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

Type: DvbSubDestinationAlignment (p. 203)
Required: False

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
Properties

**shadowColor**

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

*Type: DvbSubDestinationShadowColor (p. 207)*

*Required: False*

**fontColor**

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.

*Type: DvbSubDestinationFontColor (p. 204)*

*Required: False*

**font**

External font file used for caption burn-in. File extension must be ‘ttf’ or ‘tte’. Although the user can select output fonts for many different types of input captions, embedded, STL and teletext sources use a strict grid system. Using external fonts with these caption sources could cause unexpected display of proportional fonts. All burn-in and DVB-Sub font settings must match.

*Type: InputLocation (p. 243)*

*Required: False*

**DvbSubDestinationShadowColor**

Dvb Sub Destination Shadow Color

BLACK
NONE
WHITE

**DvbSubDestinationTeletextGridControl**

Dvb Sub Destination Teletext Grid Control

FIXED
SCALED

**DvbSubSourceSettings**

Dvb Sub Source Settings

**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

## DvbTdtSettings
**DVB Time and Date Table (SDT)**

**repInterval**

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

- **Type**: integer  
- **Required**: False  
- **Minimum**: 1000  
- **Maximum**: 30000

## Eac3AttenuationControl
**Eac3 Attenuation Control**

ATTENUATE_3_DB  
NONE

## Eac3BitstreamMode
**Eac3 Bitstream Mode**

COMMENTARY  
COMPLETE_MAIN  
EMERGENCY  
HEARING_IMPAIRED  
VISUALLY_IMPAIRED

## Eac3CodingMode
**Eac3 Coding Mode**

CODING_MODE_1_0  
CODING_MODE_2_0  
CODING_MODE_3_2

## Eac3DcFilter
**Eac3 Dc Filter**

DISABLED  
ENABLED

## Eac3DrcLine
**Eac3 Drc Line**

Eac3 Drc Line
### Properties

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

#### Eac3DrcRf

Eac3 Drc Rf
- **FILM_LIGHT**
- **FILM_STANDARD**
- **MUSIC_LIGHT**
- **MUSIC_STANDARD**
- **NONE**
- **SPEECH**

#### Eac3LfeControl

Eac3 Lfe Control
- **LFE**
- **NO_LFE**

#### Eac3LfeFilter

Eac3 Lfe Filter
- **DISABLED**
- **ENABLED**

#### Eac3MetadataControl

Eac3 Metadata Control
- **FOLLOW_INPUT**
- **USE_CONFIGURED**

#### Eac3PassthroughControl

Eac3 Passthrough Control
- **NO_PASSTHROUGH**
- **WHEN_POSSIBLE**

#### Eac3PhaseControl

Eac3 Phase Control
- **NO_SHIFT**
**SHIFT_90_DEGREES**

**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

When set to whenPossible, 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.

- **Type:** Eac3PassthroughControl (p. 209)
- **Required:** False

### drcLine

Sets the Dolby dynamic range compression profile.

- **Type:** Eac3DrcLine (p. 208)
- **Required:** False

### metadataControl

When set to followInput, 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.

- **Type:** Eac3MetadataControl (p. 209)
- **Required:** False

### bitrate

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

- **Type:** number
- **Required:** False

### ltRtSurroundMixLevel

Left total/Right total surround mix level. Only used for 3/2 coding mode.

- **Type:** number
- **Required:** False
**surroundExMode**

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

- **Type**: Eac3SurroundExMode (p. 213)
- **Required**: False

**lfeControl**

When encoding 3/2 audio, setting to lfe enables the LFE channel

- **Type**: Eac3LfeControl (p. 209)
- **Required**: False

**codingMode**

Dolby Digital Plus coding mode. Determines number of channels.

- **Type**: Eac3CodingMode (p. 208)
- **Required**: False

**surroundMode**

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

- **Type**: Eac3SurroundMode (p. 213)
- **Required**: False

**attenuationControl**

When set to attenuate3Db, applies a 3 dB attenuation to the surround channels. Only used for 3/2 coding mode.

- **Type**: Eac3AttenuationControl (p. 208)
- **Required**: False

**lfeFilter**

When set to enabled, applies a 120Hz lowpass filter to the LFE channel prior to encoding. Only valid with codingMode32 coding mode.

- **Type**: Eac3LfeFilter (p. 209)
- **Required**: False

**dcFilter**

When set to enabled, activates a DC highpass filter for all input channels.

- **Type**: Eac3DcFilter (p. 208)
- **Required**: False

**ltRtCenterMixLevel**

Left total/Right total center mix level. Only used for 3/2 coding mode.
phaseControl

When set to shift90Degrees, applies a 90-degree phase shift to the surround channels. Only used for 3/2 coding mode.

- **Type**: Eac3PhaseControl (p. 209)
- **Required**: False

bitstreamMode

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

- **Type**: Eac3BitstreamMode (p. 208)
- **Required**: False

stereoDownmix

Stereo downmix preference. Only used for 3/2 coding mode.

- **Type**: Eac3StereoDownmix (p. 212)
- **Required**: False

loRoSurroundMixLevel

Left only/Right only surround mix level. Only used for 3/2 coding mode.

- **Type**: number
- **Required**: False

drcRf

Sets the profile for heavy Dolby dynamic range compression, ensures that the instantaneous signal peaks do not exceed specified levels.

- **Type**: Eac3DrcRf (p. 209)
- **Required**: False

loRoCenterMixLevel

Left only/Right only center mix level. Only used for 3/2 coding mode.

- **Type**: number
- **Required**: False

**Eac3StereoDownmix**

Eac3 Stereo Downmix

- DPL2
- LO_RO
LT_RT
NOT_INDICATED

**Eac3SurroundExMode**

Eac3 Surround Ex Mode

- DISABLED
- ENABLED
- NOT_INDICATED

**Eac3SurroundMode**

Eac3 Surround Mode

- DISABLED
- ENABLED
- NOT_INDICATED

**EmbeddedConvert608To708**

Embedded Convert608 To708

- DISABLED
- UPCONVERT

**EmbeddedDestinationSettings**

Embedded Destination Settings

**EmbeddedPlusScte20DestinationSettings**

Embedded Plus Scte20 Destination Settings

**EmbeddedScte20Detection**

Embedded Scte20 Detection

- AUTO
- OFF

**EmbeddedSourceSettings**

Embedded Source Settings

**scte20Detection**

Set to "auto" to handle streams with intermittent and/or non-aligned SCTE-20 and Embedded captions.

*Type:* EmbeddedScte20Detection (p. 213)

*Required:* False
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

If 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.

Type: EmbeddedConvert608To708 (p. 213)
Required: False

source608TrackNumber

This field is unused and deprecated.

Type: integer
Required: False
Minimum: 1
Maximum: 5

EncoderSettings

Encoder Settings

timecodeConfig

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

Type: TimecodeConfig (p. 278)
Required: True

outputGroups

Type: Array of type OutputGroup (p. 268)
Required: True

audioDescriptions

Type: Array of type AudioDescription (p. 184)
Required: True

availConfiguration

Event-wide configuration settings for ad avail insertion.

Type: AvailConfiguration (p. 190)
Required: False
captionDescriptions
Settings for caption descriptions
  Type: Array of type CaptionDescription (p. 195)
  Required: False

globalConfiguration
Configuration settings that apply to the event as a whole.
  Type: GlobalConfiguration (p. 217)
  Required: False

videoDescriptions
  Type: Array of type VideoDescription (p. 282)
  Required: True

availBlanking
Settings for ad avail blanking.
  Type: AvailBlanking (p. 189)
  Required: False

blackoutSlate
Settings for blackout slate.
  Type: BlackoutSlate (p. 190)
  Required: False

FecOutputIncludeFec
Fec Output Include Fec
  COLUMN
  COLUMN_AND_ROW

FecOutputSettings
Fec Output Settings
rowLength
Parameter L from SMPTE 2022-1. The width of the FEC protection matrix. Must be between 1 and 20, inclusive. If only Column FEC is used, then larger values increase robustness. If Row FEC is used, then this is the number of transport stream packets per row error correction packet, and the value must be between 4 and 20, inclusive, if includeFec is columnAndRow. If includeFec is column, this value must be 1 to 20, inclusive.
  Type: integer
  Required: False
  Minimum: 1
Maximum: 20

columnDepth

Parameter D from SMPTE 2022-1. The height of the FEC protection matrix. The number of transport stream packets per column error correction packet. Must be between 4 and 20, inclusive.

Type: integer
Required: False
Minimum: 4
Maximum: 20

includeFec

Enables column only or column and row based FEC

Type: FecOutputIncludeFec (p. 215)
Required: False

FixedAfd

Fixed Afd

AFD_0000
AFD_0010
AFD_0011
AFD_0100
AFD_1000
AFD_1001
AFD_1010
AFD_1011
AFD_1101
AFD_1110
AFD_1111

FrameCaptureGroupSettings

Frame Capture Group Settings

destination

The destination for the frame capture files. Either the URI for an Amazon S3 bucket and object, plus a file name prefix (for example, s3ssl://sportsDelivery/highlights/20180820/curling_) or the URI for a MediaStore container, plus a file name prefix (for example, mediastoressl://sportsDelivery/20180820/curling_). The final file names consist of the prefix from the destination field (for example, "curling_") + name modifier + the counter (5 digits, starting from 00001) + extension (which is always .jpg). For example, curlingLow.00001.jpg

Type: OutputLocationRef (p. 269)
Required: True

FrameCaptureOutputSettings

Frame Capture Output Settings
Properties

**nameModifier**

Required if the output group contains more than one output. This modifier forms part of the output file name.

  Type: string
  Required: False

**FrameCaptureSettings**

Frame Capture Settings

**captureInterval**

The frequency, in seconds, for capturing frames for inclusion in the output. For example, "10" means capture a frame every 10 seconds.

  Type: integer
  Required: True
  Minimum: 1
  Maximum: 3600

**GatewayTimeoutException**

message

  Type: string
  Required: False

**GlobalConfiguration**

Global Configuration

**inputLossBehavior**

Settings for system actions when input is lost.

  Type: InputLossBehavior (p. 244)
  Required: False

**supportLowFramerateInputs**

Adjusts video input buffer for streams with very low video framerates. This is commonly set to enabled for music channels with less than one video frame per second.

  Type: GlobalConfigurationLowFramerateInputs (p. 218)
  Required: False

**outputLockingMode**

Indicates how MediaLive pipelines are synchronized. PIPELINELOCKING - MediaLive will attempt to synchronize the output of each pipeline to the other. EPOCHLOCKING - MediaLive will attempt to synchronize the output of each pipeline to the Unix epoch.
Properties

**Type**: GlobalConfigurationOutputLockingMode (p. 218)
**Required**: False

**initialAudioGain**
Value to set the initial audio gain for the Live Event.

**Type**: integer
**Required**: False
**Minimum**: -60
**Maximum**: 60

**inputEndAction**
Indicates the action to take when the current input completes (e.g. end-of-file). When switchAndLoopInputs is configured the encoder will restart at the beginning of the first input. When "none" is configured the encoder will transcode either black, a solid color, or a user specified slate images per the "Input Loss Behavior" configuration until the next input switch occurs (which is controlled through the Channel Schedule API).

**Type**: GlobalConfigurationInputEndAction (p. 218)
**Required**: False

**outputTimingSource**
Indicates whether the rate of frames emitted by the Live encoder should be paced by its system clock (which optionally may be locked to another source via NTP) or should be locked to the clock of the source that is providing the input stream.

**Type**: GlobalConfigurationOutputTimingSource (p. 219)
**Required**: False

**GlobalConfigurationInputEndAction**
Global Configuration Input End Action

NONE
SWITCH_AND_LOOP_INPUTS

**GlobalConfigurationLowFramerateInputs**
Global Configuration Low Framerate Inputs

DISABLED
ENABLED

**GlobalConfigurationOutputLockingMode**
Global Configuration Output Locking Mode

EPOCH_LOCKING
PIPELINE_LOCKING
GlobalConfigurationOutputTimingSource
Global Configuration Output Timing Source
  INPUT_CLOCK
  SYSTEM_CLOCK

H264AdaptiveQuantization
H264 Adaptive Quantization
  HIGH
  HIGHER
  LOW
  MAX
  MEDIUM
  OFF

H264ColorMetadata
H264 Color Metadata
  IGNORE
  INSERT

H264EntropyEncoding
H264 Entropy Encoding
  CABAC
  CAVLC

H264FlickerAq
H264 Flicker Aq
  DISABLED
  ENABLED

H264FramerateControl
H264 Framerate Control
  INITIALIZE_FROM_SOURCE
  SPECIFIED

H264GopBReference
H264 Gop BReference
  DISABLED
  ENABLED
H264GopSizeUnits

H264 Gop Size Units

FRAMES
SECONDS

H264Level

H264 Level

H264_LEVEL_1
H264_LEVEL_1_1
H264_LEVEL_1_2
H264_LEVEL_1_3
H264_LEVEL_2
H264_LEVEL_2_1
H264_LEVEL_2_2
H264_LEVEL_3
H264_LEVEL_3_1
H264_LEVEL_3_2
H264_LEVEL_4
H264_LEVEL_4_1
H264_LEVEL_4_2
H264_LEVEL_5
H264_LEVEL_5_1
H264_LEVEL_5_2
H264_LEVEL_AUTO

H264LookAheadRateControl

H264 Look Ahead Rate Control

HIGH
LOW
MEDIUM

H264ParControl

H264 Par Control

INITIALIZE_FROM_SOURCE
SPECIFIED

H264Profile

H264 Profile

BASELINE
HIGH
HIGH_10BIT
HIGH_422
HIGH_422_10BIT
MAIN
H264RateControlMode

H264 Rate Control Mode

- CBR
- QVBR
- VBR

H264ScanType

H264 Scan Type

- INTERLACED
- PROGRESSIVE

H264SceneChangeDetect

H264 Scene Change Detect

- DISABLED
- ENABLED

H264Settings

H264 Settings

minIInterval

Only meaningful if sceneChangeDetect is set to enabled. 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. Note: Maximum GOP stretch = GOP size + Min-I-interval - 1

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

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. This field is optional; when no value is specified the encoder will choose the number of slices based on encode resolution.

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

parNumerator

Pixel Aspect Ratio numerator.
Type: integer  
Required: False

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

Type: H264GopSizeUnits (p. 220)  
Required: False

subgopLength
If set to fixed, use gopNumBFrames B-frames per sub-GOP. If set to dynamic, optimize the number of B-frames used for each sub-GOP to improve visual quality.

Type: H264SubGopLength (p. 227)  
Required: False

maxBitrate
For QVBR: See the tooltip for Quality level For VBR: Set the maximum bitrate in order to accommodate expected spikes in the complexity of the video.

Type: integer  
Required: False  
Minimum: 1000

bitrate
Average bitrate in bits/second. Required when the rate control mode is VBR or CBR. Not used for QVBR. In an MS Smooth output group, each output must have a unique value when its bitrate is rounded down to the nearest multiple of 1000.

Type: integer  
Required: False  
Minimum: 1000

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

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

temporalAq
If set to enabled, adjust quantization within each frame based on temporal variation of content complexity.

Type: H264TemporalAq (p. 227)  
Required: False
afdSignaling
Indicates that AFD values will be written into the output stream. If afdSignaling is "auto", the system will try to preserve the input AFD value (in cases where multiple AFD values are valid). If set to "fixed", the AFD value will be the value configured in the fixedAfd parameter.

Type: AfdSignaling (p. 182)
Required: False

timecodeInsertion
Determines how timecodes should be inserted into the video elementary stream. - 'disabled': Do not include timecodes - 'picTimingSei': Pass through picture timing SEI messages from the source specified in Timecode Config

Type: H264TimecodeInsertionBehavior (p. 227)
Required: False

bufSize
Size of buffer (HRD buffer model) in bits/second.

Type: integer
Required: False
Minimum: 0

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

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

framerateControl
This field indicates how the output video frame rate is specified. If "specified" is selected then the output video frame rate is determined by framerateNumerator and framerateDenominator, else if "initializeFromSource" is selected then the output video frame rate will be set equal to the input video frame rate of the first input.

Type: H264FramerateControl (p. 219)
Required: False

qvbrQualityLevel
Controls the target quality for the video encode. Applies only when the rate control mode is QVBR. Set values for the QVBR quality level field and Max bitrate field that suit your most important viewing devices. Recommended values are: - Primary screen: Quality level: 8 to 10. Max bitrate: 4M - PC or tablet: Quality level: 7. Max bitrate: 1.5M to 3M - Smartphone: Quality level: 6. Max bitrate: 1M to 1.5M

Type: integer
Required: False
Minimum: 1
Maximum: 10
**fixedAfd**

Four bit AFD value to write on all frames of video in the output stream. Only valid when afdSignaling is set to 'Fixed'.

*Type:* FixedAfd (p. 216)  
*Required:* False

**level**

H.264 Level.

*Type:* H264Level (p. 220)  
*Required:* False

**lookAheadRateControl**

Amount of lookahead. A value of low can decrease latency and memory usage, while high can produce better quality for certain content.

*Type:* H264LookAheadRateControl (p. 220)  
*Required:* False

**profile**

H.264 Profile.

*Type:* H264Profile (p. 220)  
*Required:* False

**framerateNumerator**

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

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

**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

**entropyEncoding**

Entropy encoding mode. Use cabac (must be in Main or High profile) or cavlc.

*Type:* H264EntropyEncoding (p. 219)  
*Required:* False
framerateDenominator

Framerate denominator.

Type: integer
Required: False
Minimum: 1

spatialAq

If set to enabled, adjust quantization within each frame based on spatial variation of content complexity.

Type: H264SpatialAq (p. 227)
Required: False

adaptiveQuantization

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

Type: H264AdaptiveQuantization (p. 219)
Required: False

colorMetadata

Includes colorspace metadata in the output.

Type: H264ColorMetadata (p. 219)
Required: False

gopSize

GOP size (keyframe interval) in units of either frames or seconds per gopSizeUnits. Must be greater than zero.

Type: number
Required: False
Minimum: 1.0

numRefFrames

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

gopBReference

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

Type: H264GopBReference (p. 219)
Required: False
parControl

This field indicates how the output pixel aspect ratio is specified. If "specified" is selected then the output video pixel aspect ratio is determined by parNumerator and parDenominator, else if "initializeFromSource" is selected then the output pixel aspect ratio will be set equal to the input video pixel aspect ratio of the first input.

Type: H264ParControl (p. 220)
Required: False

parDenominator

Pixel Aspect Ratio denominator.

Type: integer
Required: False
Minimum: 1

syntax

Produces a bitstream compliant with SMPTE RP-2027.

Type: H264Syntax (p. 227)
Required: False

sceneChangeDetect

Scene change detection. - On: inserts I-frames when scene change is detected. - Off: does not force an I-frame when scene change is detected.

Type: H264SceneChangeDetect (p. 221)
Required: False

scanType

Sets the scan type of the output to progressive or top-field-first interlaced.

Type: H264ScanType (p. 221)
Required: False

flickerAq

If set to enabled, adjust quantization within each frame to reduce flicker or ‘pop’ on I-frames.

Type: H264FlickerAq (p. 219)
Required: False

gopNumBFrames

Number of B-frames between reference frames.

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

Rate control mode. QVBR: Quality will match the specified quality level except when it is constrained by the maximum bitrate. Recommended if you or your viewers pay for bandwidth. VBR: Quality and bitrate vary, depending on the video complexity. Recommended instead of QVBR if you want to maintain a specific average bitrate over the duration of the channel. CBR: Quality varies, depending on the video complexity. Recommended only if you distribute your assets to devices that cannot handle variable bitrates.

*Type:* H264RateControlMode (p. 221)

*Required:* False

### H264SpatialAq

H264 Spatial Aq

- DISABLED
- ENABLED

### H264SubGopLength

H264 Sub Gop Length

- DYNAMIC
- FIXED

### H264Syntax

H264 Syntax

- DEFAULT
- RP2027

### H264TemporalAq

H264 Temporal Aq

- DISABLED
- ENABLED

### H264TimecodeInsertionBehavior

H264 Timecode Insertion Behavior

- DISABLED
- PIC_TIMING_SEI

### HlsAdMarkers

Hls Ad Markers

- ADOBE
ELEMENTAL
ELEMENTAL_SCTE35

**HlsAkamaiHttpTransferMode**

Hls Akamai Http Transfer Mode

- CHUNKED
- NON_CHUNKED

**HlsAkamaiSettings**

Hls Akamai Settings

salt

Salt for authenticated Akamai.

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

httpTransferMode

Specify whether or not to use chunked transfer encoding to Akamai. User should contact Akamai to enable this feature.

- **Type**: HlsAkamaiHttpTransferMode (p. 228)
- **Required**: False

numRetries

Number of retry attempts that will be made before the Live Event is put into an error state.

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

restartDelay

If a streaming output fails, number of seconds to wait until a restart is initiated. A value of 0 means never restart.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 15

connectionRetryInterval

Number of seconds to wait before retrying connection to the CDN if the connection is lost.

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

**filecacheDuration**

Size in seconds of file cache for streaming outputs.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 600

**token**

Token parameter for authenticated akamai. If not specified, _gda_ is used.

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

**HlsBasicPutSettings**

Hls Basic Put Settings

**numRetries**

Number of retry attempts that will be made before the Live Event is put into an error state.

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

**restartDelay**

If a streaming output fails, number of seconds to wait until a restart is initiated. A value of 0 means never restart.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 15

**connectionRetryInterval**

Number of seconds to wait before retrying connection to the CDN if the connection is lost.

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

**filecacheDuration**

Size in seconds of file cache for streaming outputs.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
Maximum: 600

HlsCaptionLanguageSetting
Hls Caption Language Setting

- INSERT
- NONE
- OMIT

HlsCdnSettings
Hls Cdn Settings

HlsAkamaiSettings
Type: HlsAkamaiSettings (p. 228)
Required: False

HlsWebdavSettings
Type: HlsWebdavSettings (p. 241)
Required: False

HlsBasicPutSettings
Type: HlsBasicPutSettings (p. 229)
Required: False

HlsMediaStoreSettings
Type: HlsMediaStoreSettings (p. 238)
Required: False

HlsClientCache
Hls Client Cache

- DISABLED
- ENABLED

HlsCodecSpecification
Hls Codec Specification

- RFC_4281
- RFC_6381

HlsDirectoryStructure
Hls Directory Structure
Properties

SINGLE_DIRECTORY
SUBDIRECTORY_PER_STREAM

HlsEncryptionType

Hls Encryption Type

- AES128
- SAMPLE_AES

HlsGroupSettings

Hls Group Settings

segmentsPerSubdirectory

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

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

ivInManifest

For use with `encryptionType`. The IV (Initialization Vector) is a 128-bit number used in conjunction with the key for encrypting blocks. If set to "include", IV is listed in the manifest, otherwise the IV is not in the manifest.

- **Type**: `HlsIvInManifest` (p. 237)
- **Required**: False

outputSelection

- **Type**: `HlsOutputSelection` (p. 239)
- **Required**: False

destination

A directory or HTTP destination for the HLS segments, manifest files, and encryption keys (if enabled).

- **Type**: `OutputLocationRef` (p. 269)
- **Required**: True

encryptionType

Encrypts the segments with the given encryption scheme. Exclude this parameter if no encryption is desired.

- **Type**: `HlsEncryptionType` (p. 231)
Properties

Required: False

**indexNSegments**

Applies only if Mode field is LIVE. Specifies the maximum number of segments in the media manifest file. After this maximum, older segments are removed from the media manifest. This number must be less than or equal to the Keep Segments field.

Type: integer
Required: False
Minimum: 3

**constantIv**

For use with encryptionType. This is a 128-bit, 16-byte hex value represented by a 32-character text string. If ivSource is set to "explicit" then this parameter is required and is used as the IV for encryption.

Type: string
Required: False
MinLength: 32
MaxLength: 32

**timedMetadataId3Frame**

Indicates ID3 frame that has the timecode.

Type: HlsTimedMetadataId3Frame (p. 240)
Required: False

**baseUrlManifest**

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

**captionLanguageSetting**

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.

Type: HlsCaptionLanguageSetting (p. 230)
Required: False

**minSegmentLength**

When set, minimumSegmentLength is enforced by looking ahead and back within the specified range for a nearby avail and extending the segment size if needed.
**Properties**

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

**mode**

If "vod", all segments are indexed and kept permanently in the destination and manifest. If "live", only the number segments specified in keepSegments and indexNSegments are kept; newer segments replace older segments, which may prevent players from rewinding all the way to the beginning of the event. VOD mode uses HLS EXT-X-PLAYLIST-TYPE of EVENT while the channel is running, converting it to a "VOD" type manifest on completion of the stream.

*Type:* HlsMode (p. 239)  
*Required:* False

**keyProviderSettings**

The key provider settings.

*Type:* KeyProviderSettings (p. 248)  
*Required:* False

**manifestCompression**

When set to gzip, compresses HLS playlist.

*Type:* HlsManifestCompression (p. 238)  
*Required:* False

**ivSource**

For use with encryptionType. The IV (Initialization Vector) is a 128-bit number used in conjunction with the key for encrypting blocks. If this setting is "followsSegmentNumber", it will cause the IV to change every segment (to match the segment number). If this is set to "explicit", you must enter a constantIV value.

*Type:* HlsIvSource (p. 237)  
*Required:* False

**tsFileMode**

SEGMENTEDFILES: Emit the program as segments - multiple .ts media files. SINGLEFILE: Applies only if Mode field is VOD. Emit the program as a single .ts media file. The media manifest includes #EXT-X-BYTERANGE tags to index segments for playback. A typical use for this value is when sending the output to AWS Elemental MediaConvert, which can accept only a single media file. Playback while the channel is running is not guaranteed due to HTTP server caching.

*Type:* HlsTsFileMode (p. 241)  
*Required:* False

**manifestDurationFormat**

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

*Type:* HlsManifestDurationFormat (p. 238)
Required: False

**keyFormatVersions**

Either a single positive integer version value or a slash delimited list of version values (1/2/3).

*Type: string*

Required: False

**streamInfResolution**

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

*Type: HlsStreamInfResolution (p. 240)*

Required: False

**timestampDeltaMilliseconds**

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

*Type: integer*

Required: False

Minimum: 0

**baseUrlContent**

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

**segmentationMode**

useInputSegmentation has been deprecated. The configured segment size is always used.

*Type: HlsSegmentationMode (p. 240)*

Required: False

**captionLanguageMappings**

Mapping of up to 4 caption channels to caption languages. Is only meaningful if captionLanguageSetting is set to “insert”.

*Type: Array of type CaptionLanguageMapping (p. 197)*

Required: False

**clientCache**

When set to "disabled", sets the #EXT-X-ALLOW-CACHE:no tag in the manifest, which prevents clients from saving media segments for later replay.

*Type: HlsClientCache (p. 230)*
Required: False

codecSpecification
Specification to use (RFC-6381 or the default RFC-4281) during m3u8 playlist generation.
   Type: HlsCodecSpecification (p. 230)
   Required: False

keepSegments
Applies only if Mode field is LIVE. Specifies the number of media segments (.ts files) to retain in the destination directory.
   Type: integer
   Required: False
   Minimum: 1

redundantManifest
ENABLED: The master manifest (.m3u8 file) for each pipeline includes information about both pipelines: first its own media files, then the media files of the other pipeline. This feature allows playout device that support stale manifest detection to switch from one manifest to the other, when the current manifest seems to be stale. There are still two destinations and two master manifests, but both master manifests reference the media files from both pipelines. DISABLED: The master manifest (.m3u8 file) for each pipeline includes information about its own pipeline only. For an HLS output group with MediaPackage as the destination, the DISABLED behavior is always followed. MediaPackage regenerates the manifests it serves to players so a redundant manifest from MediaLive is irrelevant.
   Type: HlsRedundantManifest (p. 240)
   Required: False

timedMetadataId3Period
Timed Metadata interval in seconds.
   Type: integer
   Required: False
   Minimum: 0

programDateTime
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 timestampOffset.
   Type: HlsProgramDateTime (p. 240)
   Required: False

directoryStructure
Place segments in subdirectories.
   Type: HlsDirectoryStructure (p. 230)
   Required: False
keyFormat
The value specifies how the key is represented in the resource identified by the URI. If parameter is absent, an implicit value of "identity" is used. A reverse DNS string can also be given.

  Type: string
  Required: False

inputLossAction
Parameter that control output group behavior on input loss.

  Type: InputLossActionForHlsOut (p. 244)
  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 HlsAdMarkers (p. 227)
  Required: False

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

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

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: False
  Minimum: 1

hlsCdnSettings
Parameters that control interactions with the CDN.

  Type: HlsCdnSettings (p. 230)
  Required: False

iFrameOnlyPlaylists
DISABLED: Do not create an I-frame-only manifest, but do create the master and media manifests (according to the Output Selection field). STANDARD: Create an I-frame-only manifest for each output that contains video, as well as the other manifests (according to the Output Selection field). The I-frame manifest contains a #EXT-X-I-FRAMES-ONLY tag to indicate it is I-frame only, and one or more #EXT-X-BYTERANGE entries identifying the I-frame position. For example, #EXT-X-BYTERANGE:160364@1461888"
**Type**: `IFrameOnlyPlaylistType (p. 242)`

**Required**: False

---

**HlsInputSettings**

**Hls Input Settings**

**retries**

The number of consecutive times that attempts to read a manifest or segment must fail before the input is considered unavailable.

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

**bandwidth**

When specified the HLS stream with the m3u8 BANDWIDTH that most closely matches this value will be chosen, otherwise the highest bandwidth stream in the m3u8 will be chosen. The bitrate is specified in bits per second, as in an HLS manifest.

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

**retryInterval**

The number of seconds between retries when an attempt to read a manifest or segment fails.

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

**bufferSegments**

When specified, reading of the HLS input will begin this many buffer segments from the end (most recently written segment). When not specified, the HLS input will begin with the first segment specified in the m3u8.

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

---

**HlsIvInManifest**

**Hls Iv In Manifest**

**EXCLUDE**

**INCLUDE**

---

**HlsIvSource**

**Hls Iv Source**
EXPLICIT
FOLLOWS_SEGMENT_NUMBER

HlsManifestCompression
Hls Manifest Compression
  GZIP
  NONE

HlsManifestDurationFormat
Hls Manifest Duration Format
  FLOATING_POINT
  INTEGER

HlsMediaStoreSettings
Hls Media Store Settings

mediaStoreStorageClass
When set to temporal, output files are stored in non-persistent memory for faster reading and writing.
  Type: HlsMediaStoreStorageClass (p. 239)
  Required: False

numRetries
Number of retry attempts that will be made before the Live Event is put into an error state.
  Type: integer
  Required: False
  Minimum: 0

restartDelay
If a streaming output fails, number of seconds to wait until a restart is initiated. A value of 0 means never restart.
  Type: integer
  Required: False
  Minimum: 0
  Maximum: 15

connectionRetryInterval
Number of seconds to wait before retrying connection to the CDN if the connection is lost.
  Type: integer
  Required: False
  Minimum: 0
filecacheDuration
Size in seconds of file cache for streaming outputs.

  Type: integer
  Required: False
  Minimum: 0
  Maximum: 600

HlsMediaStoreStorageClass
Hls Media Store Storage Class
  TEMPORAL

HlsMode
Hls Mode
  LIVE
  VOD

HlsOutputSelection
Hls Output Selection
  MANIFESTS_AND_SEGMENTS
  SEGMENTS_ONLY

HlsOutputSettings
Hls Output Settings

  segmentModifier
String concatenated to end of segment filenames.

    Type: string
    Required: False

  hlsSettings
Settings regarding the underlying stream. These settings are different for audio-only outputs.

    Type: HlsSettings (p. 240)
    Required: True

  nameModifier
String concatenated to the end of the destination filename. Accepts "Format Identifiers ":#formatIdentifierParameters.

    Type: string
    Required: False
Properties

**MinLength**: 1

**HlsProgramDateTime**
Hls Program Date Time

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

**HlsRedundantManifest**
Hls Redundant Manifest

- **DISABLED**
- **ENABLED**

**HlsSegmentationMode**
Hls Segmentation Mode

- **USE_INPUT_SEGMENTATION**
- **USE_SEGMENT_DURATION**

**HlsSettings**
Hls Settings

**standardHlsSettings**

- **Type**: StandardHlsSettings (p. 277)
- **Required**: False

**audioOnlyHlsSettings**

- **Type**: AudioOnlyHlsSettings (p. 187)
- **Required**: False

**HlsStreamInfResolution**
Hls Stream Inf Resolution

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

**HlsTimedMetadataId3Frame**
Hls Timed Metadata Id3 Frame

- **NONE**
- **PRIV**
- **TDRL**
HlsTsFileSize

Hls Ts File Mode

- SEGMENTED_FILES
- SINGLE_FILE

HlsWebdavHttpTransferMode

Hls Webdav Http Transfer Mode

- CHUNKED
- NON CHUNKED

HlsWebdavSettings

Hls Webdav Settings

httpTransferMode

Specify whether or not to use chunked transfer encoding to WebDAV.

Type: `HlsWebdavHttpTransferMode (p. 241)`
Required: False

numRetries

Number of retry attempts that will be made before the Live Event is put into an error state.

Type: integer
Required: False
Minimum: 0

restartDelay

If a streaming output fails, number of seconds to wait until a restart is initiated. A value of 0 means never restart.

Type: integer
Required: False
Minimum: 0
Maximum: 15

connectionRetryInterval

Number of seconds to wait before retrying connection to the CDN if the connection is lost.

Type: integer
Required: False
Minimum: 0

filecacheDuration

Size in seconds of file cache for streaming outputs.
Properties

**Type**
integer
**Required**: False
**Minimum**: 0
**Maximum**: 600

**IFrameOnlyPlaylistType**
When set to "standard", an I-Frame only playlist will be written out for each video output in the output group. This I-Frame only playlist will contain byte range offsets pointing to the I-frame(s) in each segment.

- DISABLED
- STANDARD

**InputAttachment**

**inputId**
The ID of the input

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

**inputAttachmentName**
User-specified name for the attachment. This is required if the user wants to use this input in an input switch action.

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

**inputSettings**
Settings of an input (caption selector, etc.)

- **Type**: InputSettings (p. 246)
- **Required**: False

**InputChannelLevel**
Input Channel Level

**inputChannel**
The index of the input channel used as a source.

- **Type**: integer
- **Required**: True
- **Minimum**: 0
- **Maximum**: 15

**gain**
Remixing value. Units are in dB and acceptable values are within the range from -60 (mute) and 6 dB.
Type: integer  
Required: True  
Minimum: -60  
Maximum: 6

**InputCodec**

codec in increasing order of complexity

MPEG2  
AVC  
HEVC

**InputDeblockFilter**

Input Deblock Filter

DISABLED  
ENABLED

**InputDenoiseFilter**

Input Denoise Filter

DISABLED  
ENABLED

**InputFilter**

Input Filter

AUTO  
DISABLED  
FORCED

**InputLocation**

Input Location

**passwordParam**

key used to extract the password from EC2 Parameter store

Type: string  
Required: False

**uri**

Uniform Resource Identifier - This should be a path to a file accessible to the Live system (eg. a http:// URI) depending on the output type. For example, a RTMP destination should have a uri similar to: "rtmp://fmsserver/live".
username

Username if credentials are required to access a file or publishing point. This can be either a plaintext username, or a reference to an AWS parameter store name from which the username can be retrieved. AWS Parameter store format: "ssm://<parameter name>"

InputLossActionForHlsOut

Input Loss Action For Hls Out

- EMIT_OUTPUT
- PAUSE_OUTPUT

InputLossActionForMsSmoothOut

Input Loss Action For Ms Smooth Out

- EMIT_OUTPUT
- PAUSE_OUTPUT

InputLossActionForRtmpOut

Input Loss Action For Rtmp Out

- EMIT_OUTPUT
- PAUSE_OUTPUT

InputLossActionForUdpOut

Input Loss Action For Udp Out

- DROP_PROGRAM
- DROP_TS
- EMIT_PROGRAM

InputLossBehavior

Input Loss Behavior

inputLossImageType

Indicates whether to substitute a solid color or a slate into the output after input loss exceeds blackFrameMsec.

Type: InputLossImageType (p. 245)
**Properties**

**Required**: False

**inputLossImageColor**

When input loss image type is "color" this field specifies the color to use. Value: 6 hex characters representing the values of RGB.

- **Type**: string
  - **Required**: False
  - **MinLength**: 6
  - **MaxLength**: 6

**inputLossImageSlate**

When input loss image type is "slate" these fields specify the parameters for accessing the slate.

- **Type**: InputLocation (p. 243)
  - **Required**: False

**repeatFrameMsec**

On input loss, the number of milliseconds to repeat the previous picture before substituting black into the output. A value x, where 0 <= x <= 1,000,000 and a value of 1,000,000 will be interpreted as infinite.

- **Type**: integer
  - **Required**: False
  - **Minimum**: 0
  - **Maximum**: 1000000

**blackFrameMsec**

On input loss, the number of milliseconds to substitute black into the output before switching to the frame specified by inputLossImageType. A value x, where 0 <= x <= 1,000,000 and a value of 1,000,000 will be interpreted as infinite.

- **Type**: integer
  - **Required**: False
  - **Minimum**: 0
  - **Maximum**: 1000000

**InputLossImageType**

Input Loss Image Type

- COLOR
- SLATE

**InputMaximumBitrate**

Maximum input bitrate in megabits per second. Bitrates up to 50 Mbps are supported currently.

- MAX_10_MBPS
- MAX_20_MBPS
- MAX_50_MBPS
**InputResolution**

Input resolution based on lines of vertical resolution in the input; SD is less than 720 lines, HD is 720 to 1080 lines, UHD is greater than 1080 lines

- SD
- HD
- UHD

**InputSettings**

Live Event input parameters. There can be multiple inputs in a single Live Event.

**sourceEndBehavior**

Loop input if it is a file. This allows a file input to be streamed indefinitely.

- **Type:** InputSourceEndBehavior (p. 247)
- **Required:** False

**deblockFilter**

Enable or disable the deblock filter when filtering.

- **Type:** InputDeblockFilter (p. 243)
- **Required:** False

**audioSelectors**

Used to select the audio stream to decode for inputs that have multiple available.

- **Type:** Array of type AudioSelector (p. 188)
- **Required:** False

**networkInputSettings**

Input settings.

- **Type:** NetworkInputSettings (p. 266)
- **Required:** False

**inputFilter**

Turns on the filter for this input. MPEG-2 inputs have the deblocking filter enabled by default. 1) auto - filtering will be applied depending on input type/quality 2) disabled - no filtering will be applied to the input 3) forced - filtering will be applied regardless of input type

- **Type:** InputFilter (p. 243)
- **Required:** False

**videoSelector**

Informs which video elementary stream to decode for input types that have multiple available.

- **Type:** VideoSelector (p. 283)
**Required**: False

**filterStrength**
Adjusts the magnitude of filtering from 1 (minimal) to 5 (strongest).

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 5

**denoiseFilter**
Enable or disable the denoise filter when filtering.

- **Type**: InputDenoiseFilter (p. 243)
- **Required**: False

**captionSelectors**
Used to select the caption input to use for inputs that have multiple available.

- **Type**: Array of type CaptionSelector (p. 198)
- **Required**: False

**InputSourceEndBehavior**
Input Source End Behavior

- CONTINUE
- LOOP

**InputSpecification**

**codec**
Input codec

- **Type**: InputCodec (p. 243)
- **Required**: False

**resolution**
Input resolution, categorized coarsely

- **Type**: InputResolution (p. 246)
- **Required**: False

**maximumBitrate**
Maximum input bitrate, categorized coarsely

- **Type**: InputMaximumBitrate (p. 245)
- **Required**: False
**InternalServiceError**

*message*

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

**InvalidRequest**

*message*

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

**KeyProviderSettings**

Key Provider Settings

**staticKeySettings**

- **Type**: StaticKeySettings (p. 277)
- **Required**: False

**LimitExceeded**

*message*

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

**LogLevel**

The log level the user wants for their channel.

- ERROR
- WARNING
- INFO
- DEBUG
- DISABLED

**M2tsAbsentInputAudioBehavior**

M2ts Absent Input Audio Behavior

- DROP
- ENCODE_SILENCE

**M2tsArib**

M2ts Arib
DISABLED
ENABLED

**M2tsAribCaptionsPidControl**

M2ts Arib Captions Pid Control

AUTO
USE_CONFIGURED

**M2tsAudioBufferModel**

M2ts Audio Buffer Model

ATSC
DVB

**M2tsAudioInterval**

M2ts Audio Interval

VIDEO_AND_FIXED_INTERVALS
VIDEO_INTERVAL

**M2tsAudioStreamType**

M2ts Audio Stream Type

ATSC
DVB

**M2tsBufferModel**

M2ts Buffer Model

MULTIPLEX
NONE

**M2tsCcDescriptor**

M2ts Cc Descriptor

DISABLED
ENABLED

**M2tsEbifControl**

M2ts Ebif Control

NONE
PASSTHROUGH

**M2tsEbpPlacement**
M2ts Ebp Placement

- VIDEO_AND_AUDIO_PIDS
- VIDEO_PID

**M2tsEsRateInPes**
M2ts Es Rate In Pes

- EXCLUDE
- INCLUDE

**M2tsKlv**
M2ts Klv

- NONE
- PASSTHROUGH

**M2tsPcrControl**
M2ts Pcr Control

- CONFIGURED_PCR_PERIOD
- PCR_EVERY_PES_PACKET

**M2tsRateMode**
M2ts Rate Mode

- CBR
- VBR

**M2tsScte35Control**
M2ts Scte35 Control

- NONE
- PASSTHROUGH

**M2tsSegmentationMarkers**
M2ts Segmentation Markers

- EBP
- EBP_LEGACY
NONE
PSI_SEGSTART
RAI_ADAPT
RAI_SEGSTART

**M2tsSegmentationStyle**

M2ts Segmentation Style

- MAINTAIN_CADENCE
- RESET_CADENCE

**M2tsSettings**

M2ts Settings

**audioStreamType**

When set to atsc, uses stream type = 0x81 for AC3 and stream type = 0x87 for EAC3. When set to dvb, uses stream type = 0x06.

*Type: M2tsAudioStreamType (p. 249)  
Required: False*

**ecmPid**

This field is unused and deprecated.

*Type: string  
Required: False*

**dvbTeletextPid**

Packet Identifier (PID) for input source DVB Teletext data to this output. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20).8182 (or 0x1ff6).

*Type: string  
Required: False*

**aribCaptionsPidControl**

If set to auto, pid number used for ARIB Captions will be auto-selected from unused pids. If set to useConfigured, ARIB Captions will be on the configured pid number.

*Type: M2tsAribCaptionsPidControl (p. 249)  
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.

*Type: integer  
Required: False*
Properties

Minimum: 0

rateMode

When vbr, does not insert null packets into transport stream to fill specified bitrate. The bitrate setting acts as the maximum bitrate when vbr is set.

  Type: M2tsRateMode (p. 250)
  Required: False

segmentationTime

The length in seconds of each segment. Required unless markers is set to None.

  Type: number
  Required: False
  Minimum: 1.0

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. Can be entered as decimal or hexadecimal values. Each PID specified must be in the range of 32 (or 0x20).8182 (or 0x1ff6).

  Type: string
  Required: False

audioFramesPerPes

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

  Type: integer
  Required: False
  Minimum: 0

fragmentTime

The length in seconds of each fragment. Only used with EBP markers.

  Type: number
  Required: False
  Minimum: 0.0

ebpLookaheadMs

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

**ebpAudioInterval**

When `videoAndFixedIntervals` is selected, 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. Only available when EBP Cablelabs segmentation markers are selected. Partitions 1 and 2 will always follow the video interval.

Type: M2tsAudioInterval (p. 249)
Required: False

**scte35Pid**

Packet Identifier (PID) of the SCTE-35 stream in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20).8182 (or 0x1ff6).

Type: string
Required: False

**programNum**

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

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

**pmtInterval**

The number of milliseconds between instances of this table in the output transport stream. Valid values are 0, 10..1000.

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

**pcrPeriod**

Maximum time in milliseconds between Program Clock Reference (PCRs) inserted into the transport stream.

Type: integer
Required: False
Minimum: 0
Maximum: 500

**segmentationStyle**

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 "resetCadence" 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 $segmentationTime$ seconds. When a segmentation style of "maintainCadence" 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 $segmentationTime$ seconds. Note that EBP lookahead is a slight exception to this rule.

**Type:** `M2tsSegmentationStyle (p. 251)`
**Required:** False

**ebif**

If set to passthrough, passes any EBIF data from the input source to this output.

**Type:** `M2tsEbifControl (p. 249)`
**Required:** False

**audioBufferModel**

When set to dvb, uses DVB buffer model for Dolby Digital audio. When set to atsc, the ATSC model is used.

**Type:** `M2tsAudioBufferModel (p. 249)`
**Required:** False

**dvbNitSettings**

Inserts DVB Network Information Table (NIT) at the specified table repetition interval.

**Type:** `DvbNitSettings (p. 202)`
**Required:** False

**absentInputAudioBehavior**

When set to drop, output audio streams will be removed from the program if the selected input audio stream is removed from the input. This allows the output audio configuration to dynamically change based on input configuration. If this is set to encodeSilence, all output audio streams will output encoded silence when not connected to an active input stream.

**Type:** `M2tsAbsentInputAudioBehavior (p. 248)`
**Required:** False

**timedMetadataBehavior**

When set to passthrough, timed metadata will be passed through from input to output.

**Type:** `M2tsTimedMetadataBehavior (p. 258)`
**Required:** False

**timedMetadataPid**

Packet Identifier (PID) of the timed metadata stream in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

**Type:** `M2tsTimedMetadataPid (p. 258)`
**Required:** False
pmtPid

Packet Identifier (PID) for the Program Map Table (PMT) in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20) to 8182 (or 0x1ff6).

Type: string
Required: False

etvSignalPid

Packet Identifier (PID) for input source ETV Signal data to this output. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20) to 8182 (or 0x1ff6).

Type: string
Required: False

bufferModel

If set to multiplex, use multiplex buffer model for accurate interleaving. Setting to bufferModel to none can lead to lower latency, but low-memory devices may not be able to play back the stream without interruptions.

Type: M2tsBufferModel (p. 249)
Required: False

scte35Control

Optionally pass SCTE-35 signals from the input source to this output.

Type: M2tsScte35Control (p. 250)
Required: False

ebpPlacement

Controls placement of EBP on Audio PIDs. If set to videoAndAudioPids, EBP markers will be placed on the video PID and all audio PIDs. If set to videoPid, EBP markers will be placed on only the video PID.

Type: M2tsEbpPlacement (p. 250)
Required: False

arib

When set to enabled, uses ARIB-compliant field muxing and removes video descriptor.

Type: M2tsArrib (p. 248)
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
Properties

Minimum: 0.0

dvbSdtSettings
Inserts DVB Service Description Table (SDT) at the specified table repetition interval.
Type: DvbSdtSettings (p. 202)
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. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).
Type: string
Required: False

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

pcrControl
When set to pcrEveryPesPacket, 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.
Type: M2tsPcrControl (p. 250)
Required: False

videoPid
Packet Identifier (PID) of the elementary video stream in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).
Type: string
Required: False

esRateInPes
Include or exclude the ES Rate field in the PES header.
Type: M2tsEsRateInPes (p. 250)
Required: False

segmentationMarkers
Inserts segmentation markers at each segmentationTime period. raiSegstart sets the Random Access Indicator bit in the adaptation field. raiAdapt sets the RAI bit and adds the current timecode in the
private data bytes. psiSegstart 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. ebpLegacy adds Encoder Boundary Point information to the adaptation field using a legacy proprietary format.

Type: M2tsSegmentationMarkers (p. 250)
Required: False

dvbTdtSettings
Inserts DVB Time and Date Table (TDT) at the specified table repetition interval.

Type: DvbTdtSettings (p. 208)
Required: False

klv
If set to passthrough, passes any KLV data from the input source to this output.

Type: M2tsKlv (p. 250)
Required: False

ccDescriptor
When set to enabled, generates captionServiceDescriptor in PMT.

Type: M2tsCcDescriptor (p. 249)
Required: False

patInterval
The number of milliseconds between instances of this table in the output transport stream. Valid values are 0, 10..1000.

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

etvPlatformPid
Packet Identifier (PID) for input source ETV Platform data to this output. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

Type: string
Required: False

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. Can be entered as decimal or hexadecimal values. Each PID specified must be in the range of 32 (or 0x20)..8182 (or 0x1ff6).

Type: string
Required: False

aribCaptionsPid
Packet Identifier (PID) for ARIB Captions in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

  Type: string
  Required: False

scte27Pids
Packet Identifier (PID) for input source SCTE-27 data to this output. Multiple values are accepted, and can be entered in ranges and/or by comma separation. Can be entered as decimal or hexadecimal values. Each PID specified must be in the range of 32 (or 0x20)..8182 (or 0x1ff6).

  Type: string
  Required: False

klvDataPids
Packet Identifier (PID) for input source KLV data to this output. Multiple values are accepted, and can be entered in ranges and/or by comma separation. Can be entered as decimal or hexadecimal values. Each PID specified must be in the range of 32 (or 0x20)..8182 (or 0x1ff6).

  Type: string
  Required: False

M2tsTimedMetadataBehavior
M2ts Timed Metadata Behavior
  NO_PASSTHROUGH
  PASSTHROUGH

M3u8PcrControl
M3u8 Pcr Control
  CONFIGURED_PCR_PERIOD
  PCR_EVERY_PES_PACKET

M3u8Scte35Behavior
M3u8 Scte35 Behavior
  NO_PASSTHROUGH
  PASSTHROUGH

M3u8Settings
Settings information for the .m3u8 container
Properties

**pmtPid**
Packet Identifier (PID) for the Program Map Table (PMT) in the transport stream. Can be entered as a decimal or hexadecimal value.

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

**ecmPid**
This parameter is unused and deprecated.

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

**scte35Behavior**
If set to passthrough, passes any SCTE-35 signals from the input source to this output.

- **Type**: M3u8Scte35Behavior (p. 258)
- **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. Can be entered as a decimal or hexadecimal value.

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

**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. Can be entered as decimal or hexadecimal values.

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

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

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

**scte35Pid**
Packet Identifier (PID) of the SCTE-35 stream in the transport stream. Can be entered as a decimal or hexadecimal value.

- **Type**: string
Required: False

**transportStreamId**

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

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

**pcrControl**

When set to `pcrEveryPesPacket`, 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.

**Type:** M3u8PcrControl (p. 258)
**Required:** False

**videoPid**

Packet Identifier (PID) of the elementary video stream in the transport stream. Can be entered as a decimal or hexadecimal value.

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

**pcrPeriod**

Maximum time in milliseconds between Program Clock References (PCRs) inserted into the transport stream.

**Type:** integer
**Required:** False
**Minimum:** 0
**Maximum:** 500

**pmtInterval**

The number of milliseconds between instances of this table in the output transport stream. A value of "0" writes out the PMT once per segment file.

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

**programNum**

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

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

**patInterval**

The number of milliseconds between instances of this table in the output transport stream. A value of "0" writes out the PMT once per segment file.

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

**timedMetadataPid**

Packet Identifier (PID) of the timed metadata stream in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20).8182 (or 0x1ff6).

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

**timedMetadataBehavior**

When set to passthrough, timed metadata is passed through from input to output.

- **Type**: M3u8TimedMetadataBehavior (p. 261)
- **Required**: False

**M3u8TimedMetadataBehavior**

M3u8 Timed Metadata Behavior

- NO_PASSTHROUGH
- PASSTHROUGH

**MediaPackageGroupSettings**

Media Package Group Settings

**destination**

MediaPackage channel destination.

- **Type**: OutputLocationRef (p. 269)
- **Required**: True

**MediaPackageOutputDestinationSettings**

Media Package Output Destination Settings

**channelId**

ID of the channel in MediaPackage that is the destination for this output group. You do not need to specify the individual inputs in MediaPackage; MediaLive will handle the connection of the two
MediaLive pipelines to the two MediaPackage inputs. The MediaPackage channel and MediaLive channel must be in the same region.

- **Type:** string
- **Required:** False
- **MinLength:** 1

---

**MediaPackageOutputSettings**

Media Package Output Settings

**Mp2CodingMode**

Mp2 Coding Mode

- CODING_MODE_1_0
- CODING_MODE_2_0

**Mp2Settings**

Mp2 Settings

- **codingMode**
  - The MPEG2 Audio coding mode. Valid values are codingMode10 (for mono) or codingMode20 (for stereo).
  - **Type:** Mp2CodingMode (p. 262)
  - **Required:** False

- **bitrate**
  - Average bitrate in bits/second.
  - **Type:** number
  - **Required:** False

- **sampleRate**
  - Sample rate in Hz.
  - **Type:** number
  - **Required:** False

---

**MsSmoothGroupSettings**

Ms Smooth Group Settings

- **fragmentLength**
  - Length of mp4 fragments to generate (in seconds). Fragment length must be compatible with GOP size and framerate.
  - **Type:** integer
Required: False
Minimum: 1

eventId
MS Smooth event ID to be sent to the IIS server. Should only be specified if eventIdMode is set to useConfigured.

Type: string
Required: False

timestampOffset
Timestamp offset for the event. Only used if timestampOffsetMode is set to useConfiguredOffset.

Type: string
Required: False

segmentationMode
useInputSegmentation has been deprecated. The configured segment size is always used.

Type: SmoothGroupSegmentationMode (p. 276)
Required: False

numRetries
Number of retry attempts.

Type: integer
Required: False
Minimum: 0

eventStopBehavior
When set to sendEos, send EOS signal to IIS server when stopping the event.

Type: SmoothGroupEventStopBehavior (p. 276)
Required: False

acquisitionPointId
The value of the "Acquisition Point Identity" element used in each message placed in the sparse track. Only enabled if sparseTrackType is not "none".

Type: string
Required: False

sparseTrackType
If set to scte35, use incoming SCTE-35 messages to generate a sparse track in this group of MS-Smooth outputs.

Type: SmoothGroupSparseTrackType (p. 276)
Required: False
timestampOffsetMode
Type of timestamp date offset to use. - useEventStartDate: Use the date the event was started as the offset - useConfiguredOffset: Use an explicitly configured date as the offset
Type: SmoothGroupTimestampOffsetMode (p. 277)
Required: False

destination
Smooth Streaming publish point on an IIS server. Elemental Live acts as a "Push" encoder to IIS.
Type: OutputLocationRef (p. 269)
Required: True

audioOnlyTimecodeControl
If set to passthrough for an audio-only MS Smooth output, the fragment absolute time will be set to the current timecode. This option does not write timecodes to the audio elementary stream.
Type: SmoothGroupAudioOnlyTimecodeControl (p. 276)
Required: False

connectionRetryInterval
Number of seconds to wait before retrying connection to the IIS server if the connection is lost. Content will be cached during this time and the cache will be delivered to the IIS server once the connection is re-established.
Type: integer
Required: False
Minimum: 0

filecacheDuration
Size in seconds of file cache for streaming outputs.
Type: integer
Required: False
Minimum: 0

certificateMode
If set to verifyAuthenticity, verify the https certificate chain to a trusted Certificate Authority (CA). This will cause https outputs to self-signed certificates to fail.
Type: SmoothGroupCertificateMode (p. 276)
Required: False

inputLossAction
Parameter that control output group behavior on input loss.
Type: InputLossActionForMsSmoothOut (p. 244)
Required: False

**sendDelayMs**
Number of milliseconds to delay the output from the second pipeline.

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

**eventIdMode**
Specifies whether or not to send an event ID to the IIS server. If no event ID is sent and the same Live Event is used without changing the publishing point, clients might see cached video from the previous run. Options: - "useConfigured" - use the value provided in eventId - "useTimestamp" - generate and send an event ID based on the current timestamp - "noEventId" - do not send an event ID to the IIS server.

- **Type:** SmoothGroupEventIdMode (p. 276)
- **Required:** False

**restartDelay**
Number of seconds before initiating a restart due to output failure, due to exhausting the numRetries on one segment, or exceeding filecacheDuration.

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

**streamManifestBehavior**
When set to send, send stream manifest so publishing point doesn't start until all streams start.

- **Type:** SmoothGroupStreamManifestBehavior (p. 276)
- **Required:** False

**MsSmoothOutputSettings**
Ms Smooth Output Settings

**nameModifier**
String concatenated to the end of the destination filename. Required for multiple outputs of the same type.

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

**NetworkInputServerValidation**
Network Input Server Validation

- CHECK_CRYPTOGRAPHY_AND_VALIDATE_NAME
CHECK_CRYPTOGRAPHY_ONLY

**NetworkInputSettings**

Network source to transcode. Must be accessible to the Elemental Live node that is running the live event through a network connection.

**hlsInputSettings**

Specifies HLS input settings when the uri is for a HLS manifest.

- **Type**: HlsInputSettings (p. 237)
- **Required**: False

**serverValidation**

Check HTTPS server certificates. When set to checkCryptographyOnly, cryptography in the certificate will be checked, but not the server’s name. Certain subdomains (notably S3 buckets that use dots in the bucket name) do not strictly match the corresponding certificate’s wildcard pattern and would otherwise cause the event to error. This setting is ignored for protocols that do not use https.

- **Type**: NetworkInputServerValidation (p. 265)
- **Required**: False

**Output**

Output settings. There can be multiple outputs within a group.

**videoDescriptionName**

The name of the VideoDescription used as the source for this output.

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

**outputName**

The name used to identify an output.

- **Type**: string
- **Required**: False
- **MinLength**: 1
- **MaxLength**: 255

**captionDescriptionNames**

The names of the CaptionDescriptions used as caption sources for this output.

- **Type**: Array of type string
- **Required**: False

**outputSettings**

Output type-specific settings.
Properties

Type: OutputSettings (p. 269)
Required: True

_audioDescriptionNames_

The names of the AudioDescriptions used as audio sources for this output.

  Type: Array of type string
  Required: False

_OutputDestination_

_mediaPackageSettings_

Destination settings for a MediaPackage output; one destination for both encoders.

  Type: Array of type MediaPackageOutputDestinationSettings (p. 261)
  Required: False

_settings_

Destination settings for a standard output; one destination for each redundant encoder.

  Type: Array of type OutputDestinationSettings (p. 267)
  Required: False

_id_

User-specified id. This is used in an output group or an output.

  Type: string
  Required: False

_OutputDestinationSettings_

_passwordParam_

key used to extract the password from EC2 Parameter store

  Type: string
  Required: False

_streamName_

Stream name for RTMP destinations (URLs of type rtmp://)

  Type: string
  Required: False

_url_

A URL specifying a destination

  Type: string
Required: False

**username**

username for destination

Type: string
Required: False

**OutputGroup**

Output groups for this Live Event. Output groups contain information about where streams should be distributed.

**outputs**

Type: Array of type Output (p. 266)
Required: True

**outputGroupSettings**

Settings associated with the output group.

Type: OutputGroupSettings (p. 268)
Required: True

**name**

Custom output group name optionally defined by the user. Only letters, numbers, and the underscore character allowed; only 32 characters allowed.

Type: string
Required: False
MaxLength: 32

**OutputGroupSettings**

Output Group Settings

**archiveGroupSettings**

Type: ArchiveGroupSettings (p. 182)
Required: False

**mediaPackageGroupSettings**

Type: MediaPackageGroupSettings (p. 261)
Required: False

**rtmpGroupSettings**

Type: RtmpGroupSettings (p. 271)
Required: False
Properties

udpGroupSettings
Type: UdpGroupSettings (p. 279)
Required: False

msSmoothGroupSettings
Type: MsSmoothGroupSettings (p. 262)
Required: False

hlsGroupSettings
Type: HlsGroupSettings (p. 231)
Required: False

frameCaptureGroupSettings
Type: FrameCaptureGroupSettings (p. 216)
Required: False

OutputLocationRef
Reference to an OutputDestination ID defined in the channel

destinationRefId
Type: string
Required: False

OutputSettings
Output Settings

rtmpOutputSettings
Type: RtmpOutputSettings (p. 272)
Required: False

archiveOutputSettings
Type: ArchiveOutputSettings (p. 182)
Required: False

frameCaptureOutputSettings
Type: FrameCaptureOutputSettings (p. 216)
Required: False

msSmoothOutputSettings
Type: MsSmoothOutputSettings (p. 265)
Required: False
mediaPackageOutputSettings
Type: MediaPackageOutputSettings (p. 262)
Required: False

udpOutputSettings
Type: UdpOutputSettings (p. 280)
Required: False

hlsOutputSettings
Type: HlsOutputSettings (p. 239)
Required: False

PassThroughSettings
Pass Through Settings

RemixSettings
Remix Settings

channelMappings
Mapping of input channels to output channels, with appropriate gain adjustments.
Type: Array of type AudioChannelMapping (p. 183)
Required: True

channelsOut
Number of output channels to be produced. Valid values: 1, 2, 4, 6, 8
Type: integer
Required: False
Minimum: 1
Maximum: 8

channelsIn
Number of input channels to be used.
Type: integer
Required: False
Minimum: 1
Maximum: 16

ResourceConflict
message
Type: string
Required: False
ResourceNotFound

message

  Type: string
  Required: False

RtmpCacheFullBehavior

Rtmp Cache Full Behavior

  DISCONNECT_IMMEDIATELY
  WAIT_FOR_SERVER

RtmpCaptionData

Rtmp Caption Data

  ALL
  FIELD1_608
  FIELD1_AND_FIELD2_608

RtmpCaptionInfoDestinationSettings

Rtmp Caption Info Destination Settings

RtmpGroupSettings

Rtmp Group Settings

inputLossAction

Controls the behavior of this RTMP group if input becomes unavailable. - emitOutput: Emit a slate until input returns. - pauseOutput: Stop transmitting data until input returns. This does not close the underlying RTMP connection.

  Type: InputLossActionForRtmpOut (p. 244)
  Required: False

captionData

Controls the types of data that passes to onCaptionInfo outputs. If set to 'all' then 608 and 708 carried DTVCC data will be passed. If set to 'field1AndField2608' then DTVCC data will be stripped out, but 608 data from both fields will be passed. If set to 'field1608' then only the data carried in 608 from field 1 video will be passed.

  Type: RtmpCaptionData (p. 271)
  Required: False

authenticationScheme

Authentication scheme to use when connecting with CDN
**Properties**

**Type**: `AuthenticationScheme (p. 189)`  
**Required**: False

**cacheLength**

Cache length, in seconds, is used to calculate buffer size.

**Type**: integer  
**Required**: False  
**Minimum**: 30

**restartDelay**

If a streaming output fails, number of seconds to wait until a restart is initiated. A value of 0 means never restart.

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

**cacheFullBehavior**

Controls behavior when content cache fills up. If remote origin server stalls the RTMP connection and does not accept content fast enough the 'Media Cache' will fill up. When the cache reaches the duration specified by cacheLength the cache will stop accepting new content. If set to disconnectImmediately, the RTMP output will force a disconnect. Clear the media cache, and reconnect after restartDelay seconds. If set to waitForServer, the RTMP output will wait up to 5 minutes to allow the origin server to begin accepting data again.

**Type**: `RtmpCacheFullBehavior (p. 271)`  
**Required**: False

**RtmpOutputCertificateMode**

Rtmp Output Certificate Mode

- SELF_SIGNED
- VERIFY_AUTHENTICITY

**RtmpOutputSettings**

Rtmp Output Settings

**certificateMode**

If set to verifyAuthenticity, verify the tls certificate chain to a trusted Certificate Authority (CA). This will cause rtmps outputs with self-signed certificates to fail.

**Type**: `RtmpOutputCertificateMode (p. 272)`  
**Required**: False

**numRetries**

Number of retry attempts.
Type: integer  
Required: False  
Minimum: 0

destination

The RTMP endpoint excluding the stream name (e.g. rtmp://host/appname). For connection to Akamai, a username and password must be supplied. URI fields accept format identifiers.

Type: OutputLocationRef (p. 269)  
Required: True

connectionRetryInterval

Number of seconds to wait before retrying a connection to the Flash Media server if the connection is lost.

Type: integer  
Required: False  
Minimum: 1

Scte20Convert608To708

Scte20 Convert608 To708

DISABLED  
UPCONVERT

Scte20PlusEmbeddedDestinationSettings

Scte20 Plus Embedded Destination Settings

Scte20SourceSettings

Scte20 Source Settings

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

If 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.

Type: Scte20Convert608To708 (p. 273)  
Required: False
**Scte27DestinationSettings**

Scte27 Destination Settings

**Scte27SourceSettings**

Scte27 Source Settings

**pid**

The pid field is used in conjunction with the caption selector languageCode field as follows:

- Specify PID and Language: Extracts captions from that PID; the language is "informational".
- Specify PID and omit Language: Extracts the specified PID.
- Omit PID and specify Language: Extracts the specified language, whichever PID that happens to be.
- Omit PID and omit Language: Valid only if source is DVB-Sub that is being passed through; all languages will be passed through.

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

**Scte35AposNoRegionalBlackoutBehavior**

Scte35 Apos No Regional Blackout Behavior

**Scte35AposWebDeliveryAllowedBehavior**

Scte35 Apos Web Delivery Allowed Behavior

**Scte35SpliceInsert**

Scte35 Splice Insert

**adAvailOffset**

When specified, this offset (in milliseconds) is added to the input Ad Avail PTS time. This only applies to embedded SCTE 104/35 messages and does not apply to OOB messages.

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

**webDeliveryAllowedFlag**

When set to ignore, Segment Descriptors with webDeliveryAllowedFlag set to 0 will no longer trigger blackouts or Ad Avail slates
**Properties**

**Type:** `Scte35SpliceInsertWebDeliveryAllowedBehavior (p. 275)`
**Required:** False

**noRegionalBlackoutFlag**

When set to ignore, Segment Descriptors with noRegionalBlackoutFlag set to 0 will no longer trigger blackouts or Ad Avail slates

**Type:** `Scte35SpliceInsertNoRegionalBlackoutBehavior (p. 275)`
**Required:** False

**Scte35SpliceInsertNoRegionalBlackoutBehavior**

Scte35 Splice Insert No Regional Blackout Behavior

- FOLLOW
- IGNORE

**Scte35SpliceInsertWebDeliveryAllowedBehavior**

Scte35 Splice Insert Web Delivery Allowed Behavior

- FOLLOW
- IGNORE

**Scte35TimeSignalApos**

Scte35 Time Signal Apos

**adAvailOffset**

When specified, this offset (in milliseconds) is added to the input Ad Avail PTS time. This only applies to embedded SCTE 104/35 messages and does not apply to OOB messages.

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

**webDeliveryAllowedFlag**

When set to ignore, Segment Descriptors with webDeliveryAllowedFlag set to 0 will no longer trigger blackouts or Ad Avail slates

**Type:** `Scte35AposWebDeliveryAllowedBehavior (p. 274)`
**Required:** False

**noRegionalBlackoutFlag**

When set to ignore, Segment Descriptors with noRegionalBlackoutFlag set to 0 will no longer trigger blackouts or Ad Avail slates

**Type:** `Scte35AposNoRegionalBlackoutBehavior (p. 274)`
**Required**: False

**SmoothGroupAudioOnlyTimecodeControl**
Smooth Group Audio Only Timecode Control
- PASSTHROUGH
- USE_CONFIGURED_CLOCK

**SmoothGroupCertificateMode**
Smooth Group Certificate Mode
- SELF_SIGNED
- VERIFY_AUTHENTICITY

**SmoothGroupEventIdMode**
Smooth Group Event Id Mode
- NO_EVENT_ID
- USE_CONFIGURED
- USE_TIMESTAMP

**SmoothGroupEventStopBehavior**
Smooth Group Event Stop Behavior
- NONE
- SEND_EOS

**SmoothGroupSegmentationMode**
Smooth Group Segmentation Mode
- USE_INPUT_SEGMENTATION
- USE_SEGMENT_DURATION

**SmoothGroupSparseTrackType**
Smooth Group Sparse Track Type
- NONE
- SCTE_35

**SmoothGroupStreamManifestBehavior**
Smooth Group Stream Manifest Behavior
- DO_NOT_SEND
- SEND
SmoothGroupTimestampOffsetMode

Smooth Group Timestamp Offset Mode

- USE_CONFIGURED_OFFSET
- USE_EVENT_START_DATE

SmpteTtDestinationSettings

Smpte Tt Destination Settings

StandardHlsSettings

Standard Hls Settings

m3u8Settings

Type: M3u8Settings (p. 258)
Required: True

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

StaticKeySettings

Static Key Settings

staticKeyValue

Static key value as a 32 character hexadecimal string.

Type: string
Required: True
MinLength: 32
MaxLength: 32

keyProviderServer

The URL of the license server used for protecting content.

Type: InputLocation (p. 243)
Required: False

Tags

key-value pairs

Type: string
TeletextDestinationSettings

Teletext Destination Settings

TeletextSourceSettings

Teletext Source Settings

pageNumber

Specifies the teletext page number within the data stream from which to extract captions. Range of 0x100 (256) to 0x8FF (2303). Unused for passthrough. Should be specified as a hexadecimal string with no "0x" prefix.

Type: string
Required: False

TimecodeConfig

Timecode Config

syncThreshold

Threshold in frames beyond which output timecode is resynchronized to the input timecode. Discrepancies below this threshold are permitted to avoid unnecessary discontinuities in the output timecode. No timecode sync when this is not specified.

Type: integer
Required: False
Minimum: 1
Maximum: 1000000

source

Identifies the source for the timecode that will be associated with the events outputs. -Embedded (embedded): Initialize the output timecode with timecode from the the source. If no embedded timecode is detected in the source, the system falls back to using "Start at 0" (zerobased). -System Clock (systemclock): Use the UTC time. -Start at 0 (zerobased): The time of the first frame of the event will be 00:00:00:00.

Type: TimecodeConfigSource (p. 278)
Required: True

TimecodeConfigSource

Timecode Config Source

EMBEDDED
SYSTEMCLOCK
ZEROBASED

TtmlDestinationSettings

Ttml Destination Settings
styleControl
When set to passthrough, passes through style and position information from a TTML-like input source (TTML, SMPTE-TT, CFF-TT) to the CFF-TT output or TTML output.

Type: TtmlDestinationStyleControl (p. 279)
Required: False

TtmlDestinationStyleControl
Ttml Destination Style Control

PASSTHROUGH
USE_CONFIGURED

UdpContainerSettings
Udp Container Settings

m2tsSettings

Type: M2tsSettings (p. 251)
Required: False

UdpGroupSettings
Udp Group Settings

inputLossAction
Specifies behavior of last resort when input video is lost, and no more backup inputs are available. When dropTs is selected the entire transport stream will stop being emitted. When dropProgram is selected the program can be dropped from the transport stream (and replaced with null packets to meet the TS bitrate requirement). Or, when emitProgram is chosen the transport stream will continue to be produced normally with repeat frames, black frames, or slate frames substituted for the absent input video.

Type: InputLossActionForUdpOut (p. 244)
Required: False

timedMetadataId3Frame
Indicates ID3 frame that has the timecode.

Type: UdpTimedMetadataId3Frame (p. 280)
Required: False

timedMetadataId3Period
Timed Metadata interval in seconds.

Type: integer
Required: False
Minimum: 0
**UdpOutputSettings**

Udp Output Settings

**destination**

Destination address and port number for RTP or UDP packets. Can be unicast or multicast RTP or UDP (eg. rtp://239.10.10.5001 or udp://10.100.100.5002).

*Type: OutputLocationRef (p. 269)*  
*Required: True*

**bufferMsec**

UDP output buffering in milliseconds. Larger values increase latency through the transcoder but simultaneously assist the transcoder in maintaining a constant, low-jitter UDP/RTP output while accommodating clock recovery, input switching, input disruptions, picture reordering, etc.

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

**containerSettings**

*Type: UdpContainerSettings (p. 279)*  
*Required: True*

**fecOutputSettings**

Settings for enabling and adjusting Forward Error Correction on UDP outputs.

*Type: FecOutputSettings (p. 215)*  
*Required: False*

**UdpTimedMetadataId3Frame**

Udp Timed Metadata Id3 Frame

NONE  
PRIV  
TDRL

**UpdateChannel**

**inputAttachments**

*Type: Array of type InputAttachment (p. 242)*  
*Required: False*

**logLevel**

The log level to write to CloudWatch Logs.
**Properties**

**Type**

- LogLevel (p. 248)

**Required**: False

**roleArn**

An optional Amazon Resource Name (ARN) of the role to assume when running the Channel. If you do not specify this on an update call but the role was previously set that role will be removed.

- Type: string

**Required**: False

**destinations**

A list of output destinations for this channel.

- Type: Array of type OutputDestination (p. 267)

**Required**: False

**name**

The name of the channel.

- Type: string

**Required**: False

**encoderSettings**

The encoder settings for this channel.

- Type: EncoderSettings (p. 214)

**Required**: False

**inputSpecification**

Specification of input for this channel (max. bitrate, resolution, codec, etc.)

- Type: InputSpecification (p. 247)

**Required**: False

**UpdateChannelResultModel**

The updated channel's description.

**channel**

- Type: Channel (p. 199)

**Required**: False

**ValidationError**

**errorMessage**

- Type: string
**Properties**

**Required**: False

**elementPath**

*Type*: string  
*Required*: False

**VideoCodecSettings**

Video Codec Settings

**h264Settings**

*Type*: `H264Settings (p. 221)`  
*Required*: False

**frameCaptureSettings**

*Type*: `FrameCaptureSettings (p. 217)`  
*Required*: False

**VideoDescription**

Video settings for this stream.

**respondToAfd**

Indicates how to respond to the AFD values in the input stream. RESPOND causes input video to be clipped, depending on the AFD value, input display aspect ratio, and output display aspect ratio, and (except for FRAMECAPTURE codec) includes the values in the output. PASSTHROUGH (does not apply to FRAMECAPTURE codec) ignores the AFD values and includes the values in the output, so input video is not clipped. NONE ignores the AFD values and does not include the values through to the output, so input video is not clipped.

*Type*: `VideoDescriptionRespondToAfd (p. 283)`  
*Required*: False

**scalingBehavior**

STRETCHTOOUTPUT configures the output position to stretch the video to the specified output resolution (height and width). This option will override any position value. DEFAULT may insert black boxes (pillar boxes or letter boxes) around the video to provide the specified output resolution.

*Type*: `VideoDescriptionScalingBehavior (p. 283)`  
*Required*: False

**name**

The name of this VideoDescription. Outputs will use this name to uniquely identify this Description. Description names should be unique within this Live Event.

*Type*: string  
*Required*: True
**width**

Output video width, in pixels. Must be an even number. For most codecs, you can leave this field and height blank in order to use the height and width (resolution) from the source. Note, however, that leaving blank is not recommended. For the Frame Capture codec, height and width are required.

- **Type:** integer
- **Required:** False

**sharpness**

Changes the strength of the anti-alias filter used for scaling. 0 is the softest setting, 100 is the sharpest. A setting of 50 is recommended for most content.

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

**codecSettings**

Video codec settings.

- **Type:** VideoCodecSettings (p. 282)
- **Required:** False

**height**

Output video height, in pixels. Must be an even number. For most codecs, you can leave this field and width blank in order to use the height and width (resolution) from the source. Note, however, that leaving blank is not recommended. For the Frame Capture codec, height and width are required.

- **Type:** integer
- **Required:** False

**VideoDescriptionRespondToAfd**

Video Description Respond To Afd

- NONE
- PASSTHROUGH
- RESPOND

**VideoDescriptionScalingBehavior**

Video Description Scaling Behavior

- DEFAULT
- STRETCH_TO_OUTPUT

**VideoSelector**

Specifies a particular video stream within an input source. An input may have only a single video selector.
colorSpace

Specifies the colorspace of an input. This setting works in tandem with colorSpaceConversion to determine if any conversion will be performed.

Type: VideoSelectorColorSpace (p. 284)
Required: False

selectorSettings

The video selector settings.

Type: VideoSelectorSettings (p. 285)
Required: False

colorSpaceUsage

Applies only if colorSpace is a value other than follow. This field controls how the value in the colorSpace field will be used. fallback means that when the input does include color space data, that data will be used, but when the input has no color space data, the value in colorSpace will be used. Choose fallback if your input is sometimes missing color space data, but when it does have color space data, that data is correct. force means to always use the value in colorSpace. Choose force if your input usually has no color space data or might have unreliable color space data.

Type: VideoSelectorColorSpaceUsage (p. 284)
Required: False

VideoSelectorColorSpace

Video Selector Color Space

FOLLOW
REC_601
REC_709

VideoSelectorColorSpaceUsage

Video Selector Color Space Usage

FALLBACK
FORCE

VideoSelectorPid

Video Selector Pid

pid

Selects a specific PID from within a video source.

Type: integer
Required: False
Minimum: 0
Maximum: 8191
**VideoSelectorProgramId**

Video Selector Program Id

**programId**

Selects a specific program from within a multi-program transport stream. If the program doesn't exist, the first program within the transport stream will be selected by default.

*Type: integer*
*Required: False*
*Minimum: 0*
*Maximum: 65536*

**VideoSelectorSettings**

Video Selector Settings

**videoSelectorPid**

*Type: VideoSelectorPid (p. 284)*
*Required: False*

**videoSelectorProgramId**

*Type: VideoSelectorProgramId (p. 285)*
*Required: False*

**WebvttDestinationSettings**

Webvtt Destination Settings

---

### Channels channelId ChannelClass

#### URI

/prod/channels/{channelId}/channelClass

#### HTTP Methods

**PUT**

Operation ID: UpdateChannelClass

Changes the class of the channel.

#### Path Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>channelId</td>
<td>String</td>
<td>True</td>
<td>channel ID</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>UpdateChannelResultModel (p. 287)</td>
<td>The class of the channel has been successfully updated.</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 298)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 298)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>404</td>
<td>ResourceNotFound (p. 298)</td>
<td>The channel you're requesting to describe does not exist.</td>
</tr>
<tr>
<td>409</td>
<td>ResourceConflict (p. 298)</td>
<td>The channel is unable to create due to an issue with channel resources.</td>
</tr>
<tr>
<td>422</td>
<td>ChannelConfigurationValidationError</td>
<td>The Channel failed validation and could not be created.</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 299)</td>
<td>Request limit exceeded on list channel calls to channel service.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 299)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>BadGatewayException (p. 299)</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>504</td>
<td>GatewayTimeoutException (p. 299)</td>
<td>Gateway Timeout Error</td>
</tr>
</tbody>
</table>

Schemas

Request Bodies

Example PUT

```json
{
  "destinations": [
    {
      "mediaPackageSettings": [
        {
          "channelId": "string"
        }
      ],
      "settings": [
        {
          "passwordParam": "string",
          "streamName": "string",
          "url": "string",
          "username": "string"
        }
      ],
      "id": "string"
    }
  ],
  "channelClass": enum
}
```
Response Bodies

Example UpdateChannelResultModel

```json
{
  "channel": {
    "inputAttachments": [
    {
      "inputId": "string",
      "inputAttachmentName": "string",
      "inputSettings": {
        "sourceEndBehavior": enum,
        "deblockFilter": enum,
        "audioSelectors": [
          {
            "name": "string",
            "selectorSettings": {
              "audioLanguageSelection": {
                "languageSelectionPolicy": enum,
                "languageCode": "string"
              },
              "audioPidSelection": {
                "pid": integer
              }
            }
          }
        ],
        "networkInputSettings": {
          "hlsInputSettings": {
            "retries": integer,
            "bandwidth": integer,
            "retryInterval": integer,
            "bufferSegments": integer
          },
          "serverValidation": enum
        },
        "inputFilter": enum,
        "videoSelector": {
          "colorSpace": enum,
          "selectorSettings": {
            "videoSelectorPid": {
              "pid": integer
            },
            "videoSelectorProgramId": {
              "programId": integer
            }
          },
          "colorSpaceUsage": enum
        },
        "filterStrength": integer,
        "denoiseFilter": enum,
        "captionSelectors": [
        {
          "name": "string",
          "languageCode": "string",
          "selectorSettings": {
            "embeddedSourceSettings": {
              "scte20Detection": enum,
              "source608ChannelNumber": integer,
              "convert608To708": enum,
              "source608TrackNumber": integer
            },
            "scte20SourceSettings": {
              "source608ChannelNumber": integer,
            }
          }
        }
      }
    }
  }
}
```
"convert608To708": enum,
"dvbSubSourceSettings": {
  "pid": integer
},
"teletextSourceSettings": {
  "pageNumber": "string"
},
"aribSourceSettings": {
},
"scte27SourceSettings": {
  "pid": integer
}
],
"destinations": [
  {
    "mediaPackageSettings": [
      {
        "channelId": "string"
      }
    ],
    "settings": [
      {
        "passwordParam": "string",
        "streamName": "string",
        "url": "string",
        "username": "string"
      }
    ],
    "id": "string"
  }
],
"encoderSettings": {
  "timecodeConfig": {
    "syncThreshold": integer,
    "source": enum
  },
  "outputGroups": [
    {
      "outputs": [
        {
          "videoDescriptionName": "string",
          "outputName": "string",
          "captionDescriptionNames": [
            "string"
          ],
          "outputSettings": {
            "rtmpOutputSettings": {
              "certificateMode": enum,
              "numRetries": integer,
              "destination": {
                "destinationRefId": "string"
              },
              "connectionRetryInterval": integer
            },
            "archiveOutputSettings": {
              "extension": "string",
              "containerSettings": {
                "m2tsSettings": {
                  "audioStreamType": enum,
                  "ecmPId": "string",
                  "pid": integer
                }
              }
            }
          }
        }
      }
    }
  ]
}
"dvbTeletextPid": "string",
"aribCaptionsPidControl": enum,
"bitrate": integer,
"rateMode": enum,
"segmentationTime": number,
"audioPids": "string",
"audioFramesPerPes": integer,
"fragmentTime": number,
"ebpLookaheadMs": integer,
"ebpAudioInterval": enum,
"scte35Pid": "string",
"programNum": integer,
"pmtInterval": integer,
"pcrPeriod": integer,
"segmentationStyle": enum,
"ebif": enum,
"audioBufferModel": enum,
"dvbNitSettings": { 
  "networkName": "string",
  "networkId": integer,
  "repInterval": integer
},
"absentInputAudioBehavior": enum,
"timedMetadataBehavior": enum,
"timedMetadataPid": "string",
"pmtPid": "string",
"etvSignalPid": "string",
"bufferModel": enum,
"scte35Control": enum,
"ebpPlacement": enum,
"arib": enum,
"nullPacketBitrate": number,
"dvbSdtSettings": { 
  "serviceName": "string",
  "serviceProviderName": "string",
  "repInterval": integer,
  "outputSdt": enum
},
"pcrPid": "string",
"transportStreamId": integer,
"pcrControl": enum,
"videoPid": "string",
"esRateInPes": enum,
"segmentationMarkers": enum,
"dvbTdtSettings": { 
  "repInterval": integer
},
"klv": enum,
"ccDescriptor": enum,
"patInterval": integer,
"etvPlatformPid": "string",
"dvbSubPids": "string",
"aribCaptionsPid": "string",
"scte27Pids": "string",
"klvDataPids": "string"
}
Schemas

```
"udpOutputSettings": {
  "destination": {
    "destinationRefId": "string"
  },
  "bufferMsec": integer,
  "containerSettings": {
    "m2tsSettings": {
      "audioStreamType": enum,
      "ecmPid": "string",
      "dvbTeletextPid": "string",
      "aribCaptionsPidControl": enum,
      "bitrate": integer,
      "rateMode": enum,
      "segmentationTime": number,
      "audioPids": "string",
      "audioFramesPerPes": integer,
      "fragmentTime": number,
      "ebpLookaheadMs": integer,
      "ebpAudioInterval": enum,
      "scte35Pid": "string",
      "programNum": integer,
      "pmtInterval": integer,
      "pcrPeriod": integer,
      "segmentationStyle": enum,
      "ebif": enum,
      "audioBufferModel": enum,
      "dvbNitSettings": {
        "networkName": "string",
        "networkId": integer,
        "repInterval": integer
      },
      "absentInputAudioBehavior": enum,
      "timedMetadataBehavior": enum,
      "timedMetadataPid": "string",
      "pmtPid": "string",
      "etvSignalPid": "string",
      "bufferModel": enum,
      "scte35Control": enum,
      "ebpPlacement": enum,
      "arib": enum,
      "nullPacketBitrate": number,
      "dvbSdtSettings": {
        "serviceName": "string",
        "serviceProviderName": "string",
        "repInterval": integer,
        "outputSdt": enum
      },
      "pcrPid": "string",
      "transportStreamId": integer,
      "pcrControl": enum,
      "videoPid": "string",
      "esRateInPes": enum,
      "segmentationMarkers": enum,
      "dvbTdtSettings": {
        "repInterval": integer
      },
      "klv": enum,
      "ccDescriptor": enum,
      "patInterval": integer,
      "etvPlatformPid": "string",
      "dvbSubPids": "string",
      "aribCaptionsPid": "string",
      "scte27Pids": "string",
      "klvDataPids": "string"
    }
  }
}
```
"fecOutputSettings": {  
  "rowLength": integer,  
  "columnDepth": integer,  
  "includeFec": enum  
},

"hlsOutputSettings": {  
  "segmentModifier": "string",  
  "hlsSettings": {  
    "standardHlsSettings": {  
      "m3u8Settings": {  
        "pmtPid": "string",  
        "ecmPid": "string",  
        "scte35Behavior": enum,  
        "pcrPid": "string",  
        "audioPids": "string",  
        "audioFramesPerPes": integer,  
        "scte35Pid": "string",  
        "transportStreamId": integer,  
        "pcrControl": enum,  
        "videoPid": "string",  
        "pcrPeriod": integer,  
        "pmtInterval": integer,  
        "programNum": integer,  
        "patInterval": integer,  
        "timedMetadataPid": "string",  
        "timedMetadataBehavior": enum  
      },  
      "audioRenditionSets": "string"  
    },  
    "audioOnlyHlsSettings": {  
      "audioTrackType": enum,  
      "audioGroupId": "string",  
      "audioOnlyImage": {  
        "passwordParam": "string",  
        "uri": "string",  
        "username": "string"  
      }  
    }  
  },  
  "nameModifier": "string"  
},

"audioDescriptionNames": [  
  "string"  
]  
},

"outputGroupSettings": {  
  "archiveGroupSettings": {  
    "destination": {  
      "destinationRefId": "string"  
    },  
    "rolloverInterval": integer  
  },  
  "mediaPackageGroupSettings": {  
    "destination": {  
      "destinationRefId": "string"  
    }  
  },  
  "rtmpGroupSettings": {  
    "inputLossAction": enum,  
    "captionData": enum,  
    "authenticationScheme": enum,  
    "cacheLength": integer,
"restartDelay": integer,
"cacheFullBehavior": enum
},
"udpGroupSettings": {
"inputLossAction": enum,
"timedMetadataId3Frame": enum,
"timedMetadataId3Period": integer
},
"msSmoothGroupSettings": {
"fragmentLength": integer,
"eventId": "string",
"timestampOffset": "string",
"segmentationMode": enum,
"numRetries": integer,
"eventStopBehavior": enum,
"acquisitionPointId": "string",
"sparseTrackType": enum,
"timestampOffsetMode": enum,
"destination": {
"destinationRefId": "string"
},
"audioOnlyTimecodeControl": enum,
"connectionRetryInterval": integer,
"filecacheDuration": integer,
"certificateMode": enum,
"inputLossAction": enum,
"sendDelayMs": integer,
"eventIdMode": enum,
"restartDelay": integer,
"streamManifestBehavior": enum
},
"hlsGroupSettings": {
"segmentsPerSubdirectory": integer,
"ivInManifest": enum,
"outputSelection": enum,
"destination": {
"destinationRefId": "string"
},
"encryptionType": enum,
"indexNSegments": integer,
"constantIv": "string",
"timedMetadataId3Frame": enum,
"baseUrlManifest": "string",
"captionLanguageSetting": enum,
"minSegmentLength": integer,
"mode": enum,
"keyProviderSettings": {
"staticKeySettings": {
"staticKeyValue": "string",
"keyProviderServer": {
"passwordParam": "string",
"uri": "string",
"username": "string"
}
},
"manifestCompression": enum,
"ivSource": enum,
"tsFileMode": enum,
"manifestDurationFormat": enum,
"keyFormatVersions": "string",
"streamInfResolution": enum,
"timestampDeltaMilliseconds": integer,
"baseUrlContent": "string",
"segmentationMode": enum,
"captionLanguageMappings": [}
```json
{
    "languageDescription": "string",
    "captionChannel": integer,
    "languageCode": "string"
}
],
"clientCache": enum,
"codecSpecification": enum,
"keepSegments": integer,
"redundantManifest": enum,
"timedMetadataId3Period": integer,
"programDateTime": enum,
"directoryStructure": enum,
"keyFormat": "string",
"inputLossAction": enum,
"adMarkers": [
    enum
],
"programDateTimePeriod": integer,
"segmentLength": integer,
"hlsCdnSettings": {
    "hlsAkamaiSettings": {
        "salt": "string",
        "httpTransferMode": enum,
        "numRetries": integer,
        "restartDelay": integer,
        "connectionRetryInterval": integer,
        "filecacheDuration": integer,
        "token": "string"
    },
    "hlsWebdavSettings": {
        "httpTransferMode": enum,
        "numRetries": integer,
        "restartDelay": integer,
        "connectionRetryInterval": integer,
        "filecacheDuration": integer
    },
    "hlsBasicPutSettings": {
        "numRetries": integer,
        "restartDelay": integer,
        "connectionRetryInterval": integer,
        "filecacheDuration": integer
    },
    "hlsMediaStoreSettings": {
        "mediaStoreStorageClass": enum,
        "numRetries": integer,
        "restartDelay": integer,
        "connectionRetryInterval": integer,
        "filecacheDuration": integer
    }
},
"iFrameOnlyPlaylists": enum,
"frameCaptureGroupSettings": {
    "destination": {
        "destinationRefId": "string"
    }
},
"name": "string"
},
"audioDescriptions": [
    {
        "audioTypeControl": enum,
        "languageCodeControl": enum,
```
"remixSettings": {
  "channelMappings": [
    {
      "outputChannel": integer,
      "inputChannelLevels": [
        {
          "inputChannel": integer,
          "gain": integer
        }
      ]
    }
  ],
  "channelsOut": integer,
  "channelsIn": integer
},
"audioType": enum,
"name": "string",
"codecSettings": {
  "aacSettings": {
    "vbrQuality": enum,
    "codingMode": enum,
    "profile": enum,
    "inputType": enum,
    "bitrate": number,
    "rawFormat": enum,
    "rateControlMode": enum,
    "sampleRate": number,
    "spec": enum
  },
  "ac3Settings": {
    "drcProfile": enum,
    "dialnorm": integer,
    "codingMode": enum,
    "metadataControl": enum,
    "bitrate": number,
    "lfeFilter": enum,
    "bitstreamMode": enum
  },
  "eac3Settings": {
    "dialnorm": integer,
    "passthroughControl": enum,
    "drcLine": enum,
    "metadataControl": enum,
    "bitrate": number,
    "lfeFilter": enum,
    "bitstreamMode": enum,
    "surroundMode": enum,
    "lfeControl": enum,
    "codingMode": enum,
    "surroundExMode": enum,
    "metadataControl": enum,
    "bitstreamMode": enum
  },
  "passThroughSettings": {
  },
  "mp2Settings": {
    "codingMode": enum,
    "bitrate": number,
    "sampleRate": number
  }
}
"languageCode": "string",
"streamName": "string",
"audioNormalizationSettings": {
  "targetLkfs": number,
  "algorithmControl": enum,
  "algorithm": enum
},
"audioSelectorName": "string"
],
"availConfiguration": {
  "availSettings": {
    "scte35Time SIGNAL ApO": {
      "ad Avail Offset": integer,
      "web Delivery Allowed Flag": enum,
      "no Regional Blackout Flag": enum
    },
    "scte35Splice Insert": {
      "ad Avail Offset": integer,
      "web Delivery Allowed Flag": enum,
      "no Regional Blackout Flag": enum
    }
  },
  "captionDescriptions": [
    {
      "captionSelectorName": "string",
      "languageDescription": "string",
      "name": "string",
      "languageCode": "string",
      "destinationSettings": {
        "burnIn Destination Settings": {
          "xPosition": integer,
          "backgroundColor": enum,
          "yPosition": integer,
          "teletextGrid Control": enum,
          "backgroundOpacity": integer,
          "fontOpacity": integer,
          "fontResolution": integer,
          "shadowOpacity": integer,
          "shadowYOffset": integer,
          "outlineSize": integer,
          "outlineColor": enum,
          "fontColor": enum,
          "font": {
            "passwordParam": "string",
            "uri": "string",
            "username": "string"
          }
        },
        "scte27 Destination Settings": {
        },
        "teletext Destination Settings": {
        },
        "ttmlDestination Settings": {
          "style Control": enum
        },
        "smpteTt Destination Settings": {
        },
        "webvtt Destination Settings": {
        }
      }
    }
  ]
}
"embeddedPlusScte20DestinationSettings": {
"dvbSubDestinationSettings": {
"xPosition": integer,
"yPosition": integer,
"teletextGridControl": enum,
"backgroundColor": enum,
"yPosition": integer,
"teletextGridControl": enum,
"backgroundOpacity": integer,
"fontOpacity": integer,
"fontResolution": integer,
"shadowOpacity": integer,
"shadowYOffset": integer,
"outlineSize": integer,
"outlineColor": enum,
"fontSize": string,
"alignment": enum,
"shadowXOffset": integer,
"shadowColor": enum,
"fontColor": enum,
"font": {
"passwordParam": "string",
"uri": "string",
"username": "string"
},
"embeddedDestinationSettings": {
"tmpCaptionInfoDestinationSettings": {
"aribDestinationSettings": {
"scte20PlusEmbeddedDestinationSettings": {

"globalConfiguration": {
"inputLossBehavior": {
"inputLossImageType": enum,
"inputLossImageColor": "string",
"inputLossImageSlate": {
"passwordParam": "string",
"uri": "string",
"username": "string"
},
"repeatFrameMsec": integer,
"blackFrameMsec": integer
"outputLockingMode": enum,
"initialAudioGain": integer,
"inputEndAction": enum,
"outputTimingSource": enum
"videoDescriptions": [{
"respondToAfd": enum,
"scalingBehavior": enum,
"name": "string",
"width": integer,
"sharpness": integer,
"codecSettings": {
"h264Settings": {
"minIInterval": integer,
"slices": integer,"
"parNumerator": integer,
"gopSizeUnits": enum,
"subgopLength": enum,
"maxBitrate": integer,
"bitrate": integer,
"bufFillPct": integer,
"temporalAq": enum,
"afdSignaling": enum,
"timecodeInsertion": enum,
"bufSize": integer,
"softness": integer,
"framerateControl": enum,
"qvbrQualityLevel": integer,
"fixedAfd": enum,
"level": enum,
"lookAheadRateControl": enum,
"profile": enum,
"framerateNumerator": integer,
"gopClosedCadence": integer,
"entropyEncoding": enum,
"framerateDenominator": integer,
"adaptiveQuantization": enum,
"spatialAq": enum,
"colorMetadata": enum,
"gopSize": number,
"numRefFrames": integer,
"gopBReference": enum,
"parControl": enum,
"parDenominator": integer,
"syntax": enum,
"windowChangeDetect": enum,
"scanType": enum,
"flickerAq": enum,
"gopNumBFrames": integer,
"rateControlMode": enum
},
"frameCaptureSettings": {
  "captureInterval": integer
},
"height": integer
],
"availBlanking": {
  "state": enum,
  "availBlankingImage": {
    "passwordParam": "string",
    "uri": "string",
    "username": "string"
  }
},
"blackoutSlate": {
  "networkEndBlackoutImage": {
    "passwordParam": "string",
    "uri": "string",
    "username": "string"
  },
  "networkEndBlackout": enum,
  "networkId": "string",
  "state": enum,
  "blackoutSlateImage": {
    "passwordParam": "string",
    "uri": "string",
    "username": "string"
  }
}
Example InvalidRequest

{
    "message": "string"
}

Example AccessDenied

{
    "message": "string"
}

Example ResourceNotFound

{
    "message": "string"
}

Example ResourceConflict

{
    "message": "string"
}

Example ChannelConfigurationValidationError

{
    "validationErrors": [
        {
            "errorMessage": "string",
            "elementPath": "string"
        }
    ],
}
"message": "string"
}

Example LimitExceeded

{
   "message": "string"
}

Example InternalServiceError

{
   "message": "string"
}

Example BadGatewayException

{
   "message": "string"
}

Example GatewayTimeoutException

{
   "message": "string"
}

Properties

AacCodingMode

Aac Coding Mode

   AD_RECEIVER_MIX
   CODING_MODE_1_0
   CODING_MODE_1_1
   CODING_MODE_2_0
   CODING_MODE_5_1

AacInputType

Aac Input Type

   BROADCASTER_MIXED_AD
   NORMAL

AacProfile

Aac Profile
HEV1
HEV2
LC

**AacRateControlMode**

Aac Rate Control Mode

- CBR
- VBR

**AacRawFormat**

Aac Raw Format

- LATM_LOAS
- NONE

**AacSettings**

Aac Settings

**vbrQuality**

VBR Quality Level - Only used if rateControlMode is VBR.

- Type: AacVbrQuality (p. 301)
- Required: False

**codingMode**

Mono, Stereo, or 5.1 channel layout. Valid values depend on rate control mode and profile. The adReceiverMix 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.

- Type: AacCodingMode (p. 299)
- Required: False

**profile**

AAC Profile.

- Type: AacProfile (p. 299)
- Required: False

**inputType**

Set to "broadcasterMixedAd" when input contains pre-mixed main audio + AD (narration) as a stereo pair. The Audio Type field (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. The values in audioTypeControl and audioType
(in AudioDescription) are ignored when set to broadcasterMixedAd. Leave set to "normal" when input does not contain pre-mixed audio + AD.

**Type**: AacInputType (p. 299)
**Required**: False

### bitrate

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

**Type**: number
**Required**: False

### rawFormat

Sets LATM / LOAS AAC output for raw containers.

**Type**: AacRawFormat (p. 300)
**Required**: False

### rateControlMode

Rate Control Mode.

**Type**: AacRateControlMode (p. 300)
**Required**: False

### sampleRate

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

**Type**: number
**Required**: False

### spec

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

**Type**: AacSpec (p. 301)
**Required**: False

### AacSpec

Aac Spec

- MPEG2
- MPEG4

### AacVbrQuality

Aac Vbr Quality

- HIGH
Properties

LOW
MEDIUM_HIGH
MEDIUM_LOW

**Ac3BitstreamMode**

Ac3 Bitstream Mode

COMMENTARY
COMPLETE_MAIN
DIALOGUE
EMERGENCY
HEARING_IMPAIRED
MUSIC_AND_EFFECTS
VISUALLY_IMPAIRED
VOICE_OVER

**Ac3CodingMode**

Ac3 Coding Mode

CODING_MODE_1_0
CODING_MODE_1_1
CODING_MODE_2_0
CODING_MODE_3_2_LFE

**Ac3DrcProfile**

Ac3 Drc Profile

FILM_STANDARD
NONE

**Ac3LfeFilter**

Ac3 Lfe Filter

DISABLED
ENABLED

**Ac3MetadataControl**

Ac3 Metadata Control

FOLLOW_INPUT
USE_CONFIGURED

**Ac3Settings**

Ac3 Settings
**drcProfile**

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

*Type: Ac3DrcProfile (p. 302)*

*Required: False*

**dialnorm**

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

*Type: integer*

*Required: False*

*Minimum: 1*

*Maximum: 31*

**codingMode**

Dolby Digital coding mode. Determines number of channels.

*Type: Ac3CodingMode (p. 302)*

*Required: False*

**metadataControl**

When set to "followInput", 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.

*Type: Ac3MetadataControl (p. 302)*

*Required: False*

**bitrate**

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

*Type: number*

*Required: False*

**lfeFilter**

When set to enabled, applies a 120Hz lowpass filter to the LFE channel prior to encoding. Only valid in codingMode32Lfe mode.

*Type: Ac3LfeFilter (p. 302)*

*Required: False*

**bitstreamMode**

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

*Type: Ac3BitstreamMode (p. 302)*
**Properties**

**Required**: False

**AccessDenied**

**message**

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

**AfdSignaling**

Afd Signaling

- AUTO
- FIXED
- NONE

**ArchiveContainerSettings**

Archive Container Settings

**m2tsSettings**

**Type**: M2tsSettings (p. 373)
**Required**: False

**ArchiveGroupSettings**

Archive Group Settings

**destination**

A directory and base filename where archive files should be written.

**Type**: OutputLocationRef (p. 391)
**Required**: True

**rolloverInterval**

Number of seconds to write to archive file before closing and starting a new one.

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

**ArchiveOutputSettings**

Archive Output Settings

**extension**

Output file extension. If excluded, this will be auto-selected from the container type.
Properties

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

**containerSettings**

Settings specific to the container type of the file.

**Type**: ArchiveContainerSettings (p. 304)
**Required**: True

**nameModifier**

String concatenated to the end of the destination filename. Required for multiple outputs of the same type.

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

**AribDestinationSettings**

Arib Destination Settings

**AribSourceSettings**

Arib Source Settings

**AudioChannelMapping**

Audio Channel Mapping

**outputChannel**

The index of the output channel being produced.

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

**inputChannelLevels**

Indices and gain values for each input channel that should be remixed into this output channel.

**Type**: Array of type InputChannelLevel (p. 365)
**Required**: True

**AudioCodecSettings**

Audio Codec Settings

**aacSettings**

**Type**: AacSettings (p. 300)
**Required**: False
ac3Settings
  
  **Type:** Ac3Settings (p. 302)
  
  **Required:** False

eac3Settings
  
  **Type:** Eac3Settings (p. 332)
  
  **Required:** False

passThroughSettings
  
  **Type:** PassThroughSettings (p. 392)
  
  **Required:** False

mp2Settings
  
  **Type:** Mp2Settings (p. 384)
  
  **Required:** False

AudioDescription

Audio Description

audioTypeControl

Determines how audio type is determined. followInput: If the input contains an ISO 639 audioType, then that value is passed through to the output. If the input contains no ISO 639 audioType, the value in Audio Type is included in the output. useConfigured: The value in Audio Type is included in the output. Note that this field and audioType are both ignored if inputType is broadcasterMixedAd.

  
  **Type:** AudioDescriptionAudioTypeControl (p. 307)
  
  **Required:** False

languageCodeControl

Choosing followInput will cause the ISO 639 language code of the output to follow the ISO 639 language code of the input. The languageCode will be used when useConfigured is set, or when followInput is selected but there is no ISO 639 language code specified by the input.

  
  **Type:** AudioDescriptionLanguageCodeControl (p. 308)
  
  **Required:** False

remixSettings

Settings that control how input audio channels are remixed into the output audio channels.

  
  **Type:** RemixSettings (p. 392)
  
  **Required:** False

audioType

Applies only if audioTypeControl is useConfigured. The values for audioType are defined in ISO-IEC 13818-1.
### Properties

**AudioDescription**

- **Type**: `AudioType` (p. 311)
- **Required**: False

#### name

The name of this AudioDescription. Outputs will use this name to uniquely identify this AudioDescription. Description names should be unique within this Live Event.

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

#### codecSettings

Audio codec settings.

- **Type**: `AudioCodecSettings` (p. 305)
- **Required**: False

#### languageCode

Indicates the language of the audio output track. Only used if languageControlMode is `useConfigured`, or there is no ISO 639 language code specified in the input.

- **Type**: string
- **Required**: False
- **MinLength**: 3
- **MaxLength**: 3

#### streamName

Used for MS Smooth and Apple HLS outputs. Indicates the name displayed by the player (eg. English, or Director Commentary).

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

#### audioNormalizationSettings

Advanced audio normalization settings.

- **Type**: `AudioNormalizationSettings` (p. 309)
- **Required**: False

#### audioSelectorName

The name of the AudioSelector used as the source for this AudioDescription.

- **Type**: string
- **Required**: True
FOLLOW_INPUT
USE_CONFIGURED

**AudioDescriptionLanguageCodeControl**

Audio Description Language Code Control

FOLLOW_INPUT
USE_CONFIGURED

**AudioLanguageSelection**

Audio Language Selection

**languageSelectionPolicy**

When set to "strict", the transport stream demux strictly identifies audio streams by their language descriptor. If a PMT update occurs such that an audio stream matching the initially selected language is no longer present then mute will be encoded until the language returns. If "loose", then on a PMT update the demux will choose another audio stream in the program with the same stream type if it can't find one with the same language.

*Type: AudioLanguageSelectionPolicy (p. 308)*

*Required: False*

**languageCode**

Selects a specific three-letter language code from within an audio source.

*Type: string*

*Required: True*

**AudioLanguageSelectionPolicy**

Audio Language Selection Policy

LOOSE
STRICT

**AudioNormalizationAlgorithm**

Audio Normalization Algorithm

ITU_1770_1
ITU_1770_2

**AudioNormalizationAlgorithmControl**

Audio Normalization Algorithm Control

CORRECT_AUDIO
AudioNormalizationSettings

Audio Normalization Settings

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
Minimum: -59.0
Maximum: 0.0

algorithmControl

When set to correctAudio the output audio is corrected using the chosen algorithm. If set to measureOnly, the audio will be measured but not adjusted.

Type: AudioNormalizationAlgorithmControl (p. 308)
Required: False

algorithm

Audio normalization algorithm to use. itu17701 conforms to the CALM Act specification, itu17702 conforms to the EBU R-128 specification.

Type: AudioNormalizationAlgorithm (p. 308)
Required: False

AudioOnlyHlsSettings

Audio Only Hls Settings

audioTrackType

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

Type: AudioOnlyHlsTrackType (p. 310)
Required: False

audioGroupId

Specifies the group to which the audio Rendition belongs.

Type: string
Properties

**audioOnlyImage**

For use with an audio only Stream. Must be a .jpg or .png file. If given, this image will be used as the cover-art for the audio only output. Ideally, it should be formatted for an iPhone screen for two reasons. The iPhone does not resize the image, it crops a centered image on the top/bottom and left/right. Additionally, this image file gets saved bit-for-bit into every 10-second segment file, so will increase bandwidth by \( \text{image file size} \times \text{segment count} \times \text{user count}. \)

*Type:* InputLocation (p. 366)
*Required:* False

**AudioOnlyHlsTrackType**

Audio Only Hls Track Type

- ALTERNATE_AUDIO_AUTO_SELECT
- ALTERNATE_AUDIO_AUTO_SELECT_DEFAULT
- ALTERNATE_AUDIO_NOT_AUTO_SELECT
- AUDIO_ONLY_VARIANT_STREAM

**AudioPidSelection**

Audio Pid Selection

**pid**

Selects a specific PID from within a source.

*Type:* integer
*Required:* True
*Minimum:* 0
*Maximum:* 8191

**AudioSelector**

Audio Selector

**name**

The name of this AudioSelector. AudioDescriptions will use this name to uniquely identify this Selector. Selector names should be unique per input.

*Type:* string
*Required:* True
*MinLength:* 1

**selectorSettings**

The audio selector settings.

*Type:* AudioSelectorSettings (p. 311)
*Required:* False
AudioSelectorSettings

Audio Selector Settings

audioLanguageSelection

Type: AudioLanguageSelection (p. 308)
Required: False

audioPidSelection

Type: AudioPidSelection (p. 310)
Required: False

AudioType

Audio Type

CLEAN_EFFECTS
HEARING_IMPAIRED
UNDEFINED
VISUAL_IMPAIRED_COMMENTARY

AuthenticationScheme

Authentication Scheme

AKAMAI
COMMON

AvailBlanking

Avail Blanking

state

When set to enabled, causes video, audio and captions to be blanked when insertion metadata is added.

Type: AvailBlankingState (p. 311)
Required: False

availBlankingImage

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

Type: InputLocation (p. 366)
Required: False

AvailBlankingState

Avail Blanking State

DISABLED
ENABLED

**AvailConfiguration**

Avail Configuration

**availSettings**

Ad avail settings.

Type: `AvailSettings (p. 312)`

Required: False

**AvailSettings**

Avail Settings

**scte35TimeSignalApos**

Type: `Scte35TimeSignalApos (p. 398)`

Required: False

**scte35SpliceInsert**

Type: `Scte35SpliceInsert (p. 397)`

Required: False

**BadGatewayException**

**message**

Type: `string`

Required: False

**BlackoutSlate**

Blackout Slate

**networkEndBlackoutImage**

Path to local file to use as Network End Blackout image. Image will be scaled to fill the entire output raster.

Type: `InputLocation (p. 366)`

Required: False

**networkEndBlackout**

Setting to enabled causes the encoder to blackout the video, audio, and captions, and raise the "Network Blackout Image" slate when an SCTE104/35 Network End Segmentation Descriptor is encountered. The blackout will be lifted when the Network Start Segmentation Descriptor is encountered. The Network End and Network Start descriptors must contain a network ID that matches the value entered in "Network ID".
Properties

**Type**: BlackoutSlateNetworkEndBlackout (p. 313)
**Required**: False

**networkId**
Provides Network ID that matches EIDR ID format (e.g., "10.XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-C").

**Type**: string
**Required**: False
**MinLength**: 34
**MaxLength**: 34

**state**
When set to enabled, causes video, audio and captions to be blanked when indicated by program metadata.

**Type**: BlackoutSlateState (p. 313)
**Required**: False

**blackoutSlateImage**
Blackout slate image to be used. Leave empty for solid black. Only bmp and png images are supported.

**Type**: InputLocation (p. 366)
**Required**: False

**BlackoutSlateNetworkEndBlackout**
Blackout Slate Network End Blackout

**DISABLED**
**ENABLED**

**BlackoutSlateState**
Blackout Slate State

**DISABLED**
**ENABLED**

**BurnInAlignment**
Burn In Alignment

**CENTERED**
**LEFT**
**SMART**

**BurnInBackgroundColor**
Burn In Background Color
BLACK
NONE
WHITE

BurnInDestinationSettings

Burn In Destination Settings

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 xPosition is provided, the horizontal caption position will be determined by the alignment parameter. All burn-in and DVB-Sub font settings must match.

Type: integer
Required: False
Minimum: 0

backgroundColor

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

Type: BurnInBackgroundColor (p. 313)
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 yPosition is provided, the caption will be positioned towards the bottom of the output. All burn-in and DVB-Sub font settings must match.

Type: integer
Required: False
Minimum: 0

teletextGridControl

Controls whether a fixed grid size will be used to generate the output subtitles bitmap. Only applicable for Teletext inputs and DVB-Sub/Burn-in outputs.

Type: BurnInTeletextGridControl (p. 317)
Required: False

backgroundOpacity

Specifies the opacity of the background rectangle. 255 is opaque; 0 is transparent. Leaving this parameter out 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: False
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 out 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

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: 0
Maximum: 10

outlineColor

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.
Properties

**Type**: BurnInOutlineColor (p. 317)  
**Required**: False

**fontSize**

When set to 'auto' fontSize will scale depending on the size of the output. Giving a positive integer will specify the exact font size in points. All burn-in and DVB-Sub font settings must match.

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

**alignment**

If no explicit xPosition or yPosition 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. Selecting "smart" justification will left-justify live subtitles and center-justify pre-recorded subtitles. All burn-in and DVB-Sub font settings must match.

**Type**: BurnInAlignment (p. 313)  
**Required**: False

**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

**shadowColor**

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

**Type**: BurnInShadowColor (p. 317)  
**Required**: False

**fontColor**

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.

**Type**: BurnInFontColor (p. 317)  
**Required**: False

**font**

External font file used for caption burn-in. File extension must be 'ttf' or 'tte'. Although the user can select output fonts for many different types of input captions, embedded, STL and teletext sources use a strict grid system. Using external fonts with these caption sources could cause unexpected display of proportional fonts. All burn-in and DVB-Sub font settings must match.
**Properties**

**Type**: InputLocation (p. 366)
**Required**: False

**BurnInFontColor**

Burn In Font Color

BLACK
BLUE
GREEN
RED
WHITE
YELLOW

**BurnInOutlineColor**

Burn In Outline Color

BLACK
BLUE
GREEN
RED
WHITE
YELLOW

**BurnInShadowColor**

Burn In Shadow Color

BLACK
NONE
WHITE

**BurnInTeletextGridControl**

Burn In Teletext Grid Control

FIXED
SCALED

**CaptionDescription**

Output groups for this Live Event. Output groups contain information about where streams should be distributed.

**captionSelectorName**

Specifies which input caption selector to use as a caption source when generating output captions. This field should match a captionSelector name.

**Type**: string
**Required**: True
languageDescription

Human readable information to indicate captions available for players (eg. English, or Spanish).

Type: string
Required: False

name

Name of the caption description. Used to associate a caption description with an output. Names must be unique within an event.

Type: string
Required: True

languageCode


Type: string
Required: False

destinationSettings

Additional settings for captions destination that depend on the destination type.

Type: CaptionDestinationSettings (p. 318)
Required: False

CaptionDestinationSettings

Caption Destination Settings

burnInDestinationSettings

Type: BurnInDestinationSettings (p. 314)
Required: False

scte27DestinationSettings

Type: Scte27DestinationSettings (p. 396)
Required: False

teletextDestinationSettings

Type: TeletextDestinationSettings (p. 400)
Required: False

ttmlDestinationSettings

Type: TtmlDestinationSettings (p. 401)
Required: False
**CaptionLanguageMapping**

Maps a caption channel to an ISO 693-2 language code (http://www.loc.gov/standards/iso693-2), with an optional description.

**languageDescription**

Textual description of language

- **Type**: string
- **Required**: True
- **MinLength**: 1
**captionChannel**

The closed caption channel being described by this CaptionLanguageMapping. Each channel mapping must have a unique channel number (maximum of 4)

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

**languageCode**

Three character ISO 639-2 language code (see http://www.loc.gov/standards/iso639-2)

- **Type**: string
- **Required**: True
- **MinLength**: 3
- **MaxLength**: 3

**CaptionSelector**

Output groups for this Live Event. Output groups contain information about where streams should be distributed.

**name**

Name identifier for a caption selector. This name is used to associate this caption selector with one or more caption descriptions. Names must be unique within an event.

- **Type**: string
- **Required**: True
- **MinLength**: 1

**languageCode**

When specified this field indicates the three letter language code of the caption track to extract from the source.

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

**selectorSettings**

Caption selector settings.

- **Type**: CaptionSelectorSettings (p. 320)
- **Required**: False

**CaptionSelectorSettings**

Caption Selector Settings

**embeddedSourceSettings**

- **Type**: EmbeddedSourceSettings (p. 335)
Required: False

**scte20SourceSettings**

Type: [Scte20SourceSettings](p. 396)
Required: False

**dvbSubSourceSettings**

Type: [DvbSubSourceSettings](p. 329)
Required: False

**teletextSourceSettings**

Type: [TeletextSourceSettings](p. 400)
Required: False

**aribSourceSettings**

Type: [AribSourceSettings](p. 305)
Required: False

**scte27SourceSettings**

Type: [Scte27SourceSettings](p. 396)
Required: False

**Channel**

**inputAttachments**

List of input attachments for channel.

Type: Array of type [InputAttachment](p. 364)
Required: False

**destinations**

A list of destinations of the channel. For UDP outputs, there is one destination per output. For other types (HLS, for example), there is one destination per packager.

Type: Array of type [OutputDestination](p. 389)
Required: False

**encoderSettings**

Type: [EncoderSettings](p. 336)
Required: False

**egressEndpoints**

The endpoints where outgoing connections initiate from
Properties

- **Type**: Array of type ChannelEgressEndpoint (p. 323)
  - **Required**: False

- **inputSpecification**
  - **Type**: InputSpecification (p. 370)
  - **Required**: False

- **channelClass**
  - The class for this channel. STANDARD for a channel with two pipelines or SINGLE_PIPELINE for a channel with one pipeline.
  - **Type**: ChannelClass (p. 323)
  - **Required**: False

- **tags**
  - A collection of key-value pairs.
  - **Type**: Tags (p. 400)
  - **Required**: False

- **logLevel**
  - The log level being written to CloudWatch Logs.
  - **Type**: LogLevel (p. 371)
  - **Required**: False

- **roleArn**
  - The Amazon Resource Name (ARN) of the role assumed when running the Channel.
  - **Type**: string
  - **Required**: False

- **name**
  - The name of the channel. (user-mutable)
  - **Type**: string
  - **Required**: False

- **id**
  - The unique id of the channel.
  - **Type**: string
  - **Required**: False

- **state**
  - **Type**: ChannelState (p. 323)
Required: False

pipelinesRunningCount
The number of currently healthy pipelines.
  Type: integer
  Required: False

arn
The unique arn of the channel.
  Type: string
  Required: False

ChannelClass
A standard channel has two encoding pipelines and a single pipeline channel only has one.
  STANDARD
  SINGLE_PIPELINE

ChannelConfigurationValidationError
validationErrors
A collection of validation error responses.
  Type: Array of type ValidationError (p. 403)
  Required: False

message
  Type: string
  Required: False

ChannelEgressEndpoint
sourceIp
Public IP of where a channel's output comes from
  Type: string
  Required: False

ChannelState
  CREATING
  CREATE_FAILED
  IDLE
STARTING
RUNNING
RECOVERING
STOPPING
DELETING
DELETED
UPDATING
UPDATE_FAILED

**DvbNitSettings**

DVB Network Information Table (NIT)

**networkName**

The network name text placed in the networkNameDescriptor inside the Network Information Table. Maximum length is 256 characters.

- **Type:** string
- **Required:** True
- **MinLength:** 1
- **MaxLength:** 256

**networkId**

The numeric value placed in the Network Information Table (NIT).

- **Type:** integer
- **Required:** True
- **Minimum:** 0
- **Maximum:** 65536

**repInterval**

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

- **Type:** integer
- **Required:** False
- **Minimum:** 25
- **Maximum:** 10000

**DvbSdtOutputSdt**

Dvb Sdt Output Sdt

- **SDT_FOLLOW**
- **SDT_FOLLOW_IF_PRESENT**
- **SDT_MANUAL**
- **SDT_NONE**

**DvbSdtSettings**

DVB Service Description Table (SDT)
**serviceName**

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

- **Type:** string
- **Required:** False
- **MinLength:** 1
- **MaxLength:** 256

**serviceProviderName**

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

- **Type:** string
- **Required:** False
- **MinLength:** 1
- **MaxLength:** 256

**repInterval**

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

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

**outputSdt**

Selects method of inserting SDT information into output stream. The sdtFollow setting copies SDT information from input stream to output stream. The sdtFollowIfPresent setting 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. The sdtManual setting means user will enter the SDT information. The sdtNone setting means output stream will not contain SDT information.

- **Type:** DvbSdtOutputSdt (p. 324)
- **Required:** False

**DvbSubDestinationAlignment**

Dvb Sub Destination Alignment

- CENTERED
- LEFT
- SMART

**DvbSubDestinationBackgroundColor**

Dvb Sub Destination Background Color

- BLACK
- NONE
DvbSubDestinationFontColor

Dvb Sub Destination Font Color

BLACK  
BLUE  
GREEN  
RED  
WHITE  
YELLOW

DvbSubDestinationOutlineColor

Dvb Sub Destination Outline Color

BLACK  
BLUE  
GREEN  
RED  
WHITE  
YELLOW

DvbSubDestinationSettings

Dvb Sub Destination Settings

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 xPosition 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

backgroundColor

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

Type: DvbSubDestinationBackgroundColor (p. 325)  
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 yPosition 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

### teletextGridControl

Controls whether a fixed grid size will be used to generate the output subtitles bitmap. Only applicable for Teletext inputs and DVB-Sub/Burn-in outputs.

**Type**: DvbSubDestinationTeletextGridControl (p. 329)  
**Required**: False

### 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**: False  
**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

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: 0
Maximum: 10

outlineColor

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.

Type: DvbSubDestinationOutlineColor (p. 326)
Required: False

fontSize

When set to auto fontSize will scale depending on the size of the output. Giving a positive integer will specify the exact font size in points. All burn-in and DVB-Sub font settings must match.

Type: string
Required: False

alignment

If no explicit xPosition or yPosition 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. Selecting "smart" justification will left-justify live subtitles and center-justify pre-recorded subtitles. This option is not valid for source captions that are STL or 608/embedded. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

Type: DvbSubDestinationAlignment (p. 325)
Required: False

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
**shadowColor**

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

*Type:* DvbSubDestinationShadowColor (p. 329)
*Required:* False

**fontColor**

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.

*Type:* DvbSubDestinationFontColor (p. 326)
*Required:* False

**font**

External font file used for caption burn-in. File extension must be ‘ttf’ or ‘tte’. Although the user can select output fonts for many different types of input captions, embedded, STL and teletext sources use a strict grid system. Using external fonts with these caption sources could cause unexpected display of proportional fonts. All burn-in and DVB-Sub font settings must match.

*Type:* InputLocation (p. 366)
*Required:* False

**DvbSubDestinationShadowColor**

Dvb Sub Destination Shadow Color

- BLACK
- NONE
- WHITE

**DvbSubDestinationTeletextGridControl**

Dvb Sub Destination Teletext Grid Control

- FIXED
- SCALED

**DvbSubSourceSettings**

Dvb Sub Source Settings

**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
**Properties**

**Required**: False
**Minimum**: 1

**DvbTdtSettings**

DVB Time and Date Table (SDT)

**repInterval**

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

**Type**: integer
**Required**: False
**Minimum**: 1000
**Maximum**: 30000

**Eac3AttenuationControl**

Eac3 Attenuation Control

ATTENUATE_3_DB
NONE

**Eac3BitstreamMode**

Eac3 Bitstream Mode

COMMENTARY
COMPLETE_MAIN
EMERGENCY
HEARING_IMPAIRED
VISUALLY_IMPAIRED

**Eac3CodingMode**

Eac3 Coding Mode

CODING_MODE_1_0
CODING_MODE_2_0
CODING_MODE_3_2

**Eac3DcFilter**

Eac3 Dc Filter

DISABLED
ENABLED

**Eac3DrcLine**

Eac3 Drc Line
FILM_LIGHT
FILM_STANDARD
MUSIC_LIGHT
MUSIC_STANDARD
NONE
SPEECH

**Eac3DrcRf**

Eac3 Drc Rf

FILM_LIGHT
FILM_STANDARD
MUSIC_LIGHT
MUSIC_STANDARD
NONE
SPEECH

**Eac3LfeControl**

Eac3 Lfe Control

LFE
NO_LFE

**Eac3LfeFilter**

Eac3 Lfe Filter

DISABLED
ENABLED

**Eac3MetadataControl**

Eac3 Metadata Control

FOLLOW_INPUT
USE_CONFIGURED

**Eac3PassthroughControl**

Eac3 Passthrough Control

NO_PASSTHROUGH
WHEN_POSSIBLE

**Eac3PhaseControl**

Eac3 Phase Control

NO_SHIFT
### Eac3Settings

Eac3 Settings

#### 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

When set to whenPossible, 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.

- **Type**: [Eac3PassthroughControl](p. 331)
- **Required**: False

#### drcLine

Sets the Dolby dynamic range compression profile.

- **Type**: [Eac3DrcLine](p. 330)
- **Required**: False

#### metadataControl

When set to followInput, 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.

- **Type**: [Eac3MetadataControl](p. 331)
- **Required**: False

#### bitrate

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

- **Type**: number
- **Required**: False

#### ltRtSurroundMixLevel

Left total/Right total surround mix level. Only used for 3/2 coding mode.

- **Type**: number
- **Required**: False
**surroundExMode**

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

*Type:* Eac3SurroundExMode (p. 335)  
*Required:* False

**lfeControl**

When encoding 3/2 audio, setting to lfe enables the LFE channel

*Type:* Eac3LfeControl (p. 331)  
*Required:* False

**codingMode**

Dolby Digital Plus coding mode. Determines number of channels.

*Type:* Eac3CodingMode (p. 330)  
*Required:* False

**surroundMode**

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

*Type:* Eac3SurroundMode (p. 335)  
*Required:* False

**attenuationControl**

When set to attenuate3Db, applies a 3 dB attenuation to the surround channels. Only used for 3/2 coding mode.

*Type:* Eac3AttenuationControl (p. 330)  
*Required:* False

**lfeFilter**

When set to enabled, applies a 120Hz lowpass filter to the LFE channel prior to encoding. Only valid with codingMode32 coding mode.

*Type:* Eac3LfeFilter (p. 331)  
*Required:* False

**dcFilter**

When set to enabled, activates a DC highpass filter for all input channels.

*Type:* Eac3DcFilter (p. 330)  
*Required:* False

**ltRtCenterMixLevel**

Left total/Right total center mix level. Only used for 3/2 coding mode.
Properties

**phaseControl**
When set to shift90Degrees, applies a 90-degree phase shift to the surround channels. Only used for 3/2 coding mode.

**Type**: Eac3PhaseControl (p. 331)
**Required**: False

**bitstreamMode**
 Specifies the bitstream mode (bsmod) for the emitted E-AC-3 stream. See ATSC A/52-2012 (Annex E) for background on these values.

**Type**: Eac3BitstreamMode (p. 330)
**Required**: False

**stereoDownmix**
Stereo downmix preference. Only used for 3/2 coding mode.

**Type**: Eac3StereoDownmix (p. 334)
**Required**: False

**loRoSurroundMixLevel**
Left only/Right only surround mix level. Only used for 3/2 coding mode.

**Type**: number
**Required**: False

**drcRf**
Sets the profile for heavy Dolby dynamic range compression, ensures that the instantaneous signal peaks do not exceed specified levels.

**Type**: Eac3DrcRf (p. 331)
**Required**: False

**loRoCenterMixLevel**
Left only/Right only center mix level. Only used for 3/2 coding mode.

**Type**: number
**Required**: False

**Eac3StereoDownmix**
Eac3 Stereo Downmix

DPL2
LO_RO
LT_RT
NOT_INDICATED

**Eac3SurroundExMode**

Eac3 Surround Ex Mode

- DISABLED
- ENABLED
- NOT_INDICATED

**Eac3SurroundMode**

Eac3 Surround Mode

- DISABLED
- ENABLED
- NOT_INDICATED

**EmbeddedConvert608To708**

Embedded Convert608 To708

- DISABLED
- UPCONVERT

**EmbeddedDestinationSettings**

Embedded Destination Settings

**EmbeddedPlusScte20DestinationSettings**

Embedded Plus Scte20 Destination Settings

**EmbeddedScte20Detection**

Embedded Scte20 Detection

- AUTO
- OFF

**EmbeddedSourceSettings**

Embedded Source Settings

**scte20Detection**

Set to "auto" to handle streams with intermittent and/or non-aligned SCTE-20 and Embedded captions.

*Type: EmbeddedScte20Detection (p. 335)*
Properties

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

If 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.

Type: EmbeddedConvert608To708 (p. 335)
Required: False

source608TrackNumber

This field is unused and deprecated.

Type: integer
Required: False
Minimum: 1
Maximum: 5

EncoderSettings

Encoder Settings

timecodeConfig

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

Type: TimecodeConfig (p. 400)
Required: True

outputGroups

Type: Array of type OutputGroup (p. 390)
Required: True

audioDescriptions

Type: Array of type AudioDescription (p. 306)
Required: True

availConfiguration

Event-wide configuration settings for ad avail insertion.
Properties

Type: AvailConfiguration (p. 312)
Required: False

captionDescriptions

Settings for caption descriptions

Type: Array of type CaptionDescription (p. 317)
Required: False

globalConfiguration

Configuration settings that apply to the event as a whole.

Type: GlobalConfiguration (p. 339)
Required: False

videoDescriptions

Type: Array of type VideoDescription (p. 404)
Required: True

availBlanking

Settings for ad avail blanking.

Type: AvailBlanking (p. 311)
Required: False

blackoutSlate

Settings for blackout slate.

Type: BlackoutSlate (p. 312)
Required: False

FecOutputIncludeFec

Fec Output Include Fec

COLUMN
COLUMN_AND_ROW

FecOutputSettings

Fec Output Settings

rowLength

Parameter L from SMPTE 2022-1. The width of the FEC protection matrix. Must be between 1 and 20, inclusive. If only Column FEC is used, then larger values increase robustness. If Row FEC is used, then this is the number of transport stream packets per row error correction packet, and the value must be
between 4 and 20, inclusive, if includeFec is columnAndRow. If includeFec is column, this value must be 1 to 20, inclusive.

    Type: integer
    Required: False
    Minimum: 1
    Maximum: 20

**columnDepth**

Parameter D from SMPTE 2022-1. The height of the FEC protection matrix. The number of transport stream packets per column error correction packet. Must be between 4 and 20, inclusive.

    Type: integer
    Required: False
    Minimum: 4
    Maximum: 20

**includeFec**

Enables column only or column and row based FEC

    Type: FecOutputIncludeFec (p. 337)
    Required: False

**FixedAfd**

Fixed Afd

    AFD_0000
    AFD_0010
    AFD_0011
    AFD_0100
    AFD_1000
    AFD_1001
    AFD_1010
    AFD_1011
    AFD_1101
    AFD_1110
    AFD_1111

**FrameCaptureGroupSettings**

Frame Capture Group Settings

**destination**

The destination for the frame capture files. Either the URI for an Amazon S3 bucket and object, plus a file name prefix (for example, s3ssl://sportsDelivery/highlights/20180820/curling_) or the URI for a MediaStore container, plus a file name prefix (for example, mediastoressl://sportsDelivery/20180820/curling_). The final file names consist of the prefix from the destination field (for example, "curling_") + name modifier + the counter (5 digits, starting from 00001) + extension (which is always .jpg). For example, curlingLow.00001.jpg
**Type**: OutputLocationRef (p. 391)  
**Required**: True

**FrameCaptureOutputSettings**

Frame Capture Output Settings

**nameModifier**

Required if the output group contains more than one output. This modifier forms part of the output file name.

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

**FrameCaptureSettings**

Frame Capture Settings

**captureInterval**

The frequency, in seconds, for capturing frames for inclusion in the output. For example, "10" means capture a frame every 10 seconds.

**Type**: integer  
**Required**: True  
**Minimum**: 1  
**Maximum**: 3600

**GatewayTimeoutException**

**message**

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

**GlobalConfiguration**

Global Configuration

**inputLossBehavior**

Settings for system actions when input is lost.

**Type**: InputLossBehavior (p. 367)  
**Required**: False

**supportLowFramerateInputs**

Adjusts video input buffer for streams with very low video framerates. This is commonly set to enabled for music channels with less than one video frame per second.

**Type**: GlobalConfigurationLowFramerateInputs (p. 340)
Properties

**outputLockingMode**

Indicates how MediaLive pipelines are synchronized. PIPELINELOCKING - MediaLive will attempt to synchronize the output of each pipeline to the other. EPOCHLOCKING - MediaLive will attempt to synchronize the output of each pipeline to the Unix epoch.

*Type:* GlobalConfigurationOutputLockingMode (p. 341)
*Required:* False

**initialAudioGain**

Value to set the initial audio gain for the Live Event.

*Type:* integer
*Required:* False
*Minimum:* -60
*Maximum:* 60

**inputEndAction**

Indicates the action to take when the current input completes (e.g. end-of-file). When switchAndLoopInputs is configured the encoder will restart at the beginning of the first input. When "none" is configured the encoder will transcode either black, a solid color, or a user specified slate images per the "Input Loss Behavior" configuration until the next input switch occurs (which is controlled through the Channel Schedule API).

*Type:* GlobalConfigurationInputEndAction (p. 340)
*Required:* False

**outputTimingSource**

Indicates whether the rate of frames emitted by the Live encoder should be paced by its system clock (which optionally may be locked to another source via NTP) or should be locked to the clock of the source that is providing the input stream.

*Type:* GlobalConfigurationOutputTimingSource (p. 341)
*Required:* False

**GlobalConfigurationInputEndAction**

Global Configuration Input End Action

NONE
SWITCH_AND_LOOP_INPUTS

**GlobalConfigurationLowFramerateInputs**

Global Configuration Low Framerate Inputs

DISABLED
ENABLED
**GlobalConfigurationOutputLockingMode**

Global Configuration Output Locking Mode

- EPOCH_LOCKING
- PIPELINE_LOCKING

**GlobalConfigurationOutputTimingSource**

Global Configuration Output Timing Source

- INPUT_CLOCK
- SYSTEM_CLOCK

**H264AdaptiveQuantization**

H264 Adaptive Quantization

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

**H264ColorMetadata**

H264 Color Metadata

- IGNORE
- INSERT

**H264EntropyEncoding**

H264 Entropy Encoding

- CABAC
- CAVLC

**H264FlickerAq**

H264 Flicker Aq

- DISABLED
- ENABLED

**H264FramerateControl**

H264 Framerate Control
INITIALIZE_FROM_SOURCE
SPECIFIED

H264GopBReference

H264 Gop BReference
  DISABLED
  ENABLED

H264GopSizeUnits

H264 Gop Size Units
  FRAMES
  SECONDS

H264Level

H264 Level
  H264_LEVEL_1
  H264_LEVEL_1_1
  H264_LEVEL_1_2
  H264_LEVEL_1_3
  H264_LEVEL_2
  H264_LEVEL_2_1
  H264_LEVEL_2_2
  H264_LEVEL_3
  H264_LEVEL_3_1
  H264_LEVEL_3_2
  H264_LEVEL_4
  H264_LEVEL_4_1
  H264_LEVEL_4_2
  H264_LEVEL_5
  H264_LEVEL_5_1
  H264_LEVEL_5_2
  H264_LEVEL_AUTO

H264LookAheadRateControl

H264 Look Ahead Rate Control
  HIGH
  LOW
  MEDIUM

H264ParControl

H264 Par Control
INITIALIZE_FROM_SOURCE
SPECIFIED

H264Profile
H264 Profile
BASELINE
HIGH
HIGH_10BIT
HIGH_422
HIGH_422_10BIT
MAIN

H264RateControlMode
H264 Rate Control Mode
CBR
QVBR
VBR

H264ScanType
H264 Scan Type
INTERLACED
PROGRESSIVE

H264SceneChangeDetect
H264 Scene Change Detect
DISABLED
ENABLED

H264Settings
H264 Settings

minIInterval

Only meaningful if sceneChangeDetect is set to enabled. 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. Note: Maximum GOP stretch = GOP size + Min-I-interval - 1

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

**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. This field is optional; when no value is specified the encoder will choose the number of slices based on encode resolution.

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

**parNumerator**

Pixel Aspect Ratio numerator.

**Type:** integer  
**Required:** False

**gopSizeUnits**

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

**Type:** H264GopSizeUnits (p. 342)  
**Required:** False

**subgopLength**

If set to fixed, use gopNumBFrames B-frames per sub-GOP. If set to dynamic, optimize the number of B-frames used for each sub-GOP to improve visual quality.

**Type:** H264SubGopLength (p. 349)  
**Required:** False

**maxBitrate**

For QVBR: See the tooltip for Quality level For VBR: Set the maximum bitrate in order to accommodate expected spikes in the complexity of the video.

**Type:** integer  
**Required:** False  
**Minimum:** 1000

**bitrate**

Average bitrate in bits/second. Required when the rate control mode is VBR or CBR. Not used for QVBR. In an MS Smooth output group, each output must have a unique value when its bitrate is rounded down to the nearest multiple of 1000.

**Type:** integer  
**Required:** False
Minimum: 1000

**bufFillPct**

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

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

**temporalAq**

If set to enabled, adjust quantization within each frame based on temporal variation of content complexity.

- **Type:** [H264TemporalAq](p. 350)
- **Required:** False

**afdSignaling**

Indicates that AFD values will be written into the output stream. If afdSignaling is "auto", the system will try to preserve the input AFD value (in cases where multiple AFD values are valid). If set to "fixed", the AFD value will be the value configured in the fixedAfd parameter.

- **Type:** [AfdSignaling](p. 304)
- **Required:** False

**timecodeInsertion**

Determines how timecodes should be inserted into the video elementary stream. - 'disabled': Do not include timecodes - 'picTimingSei': Pass through picture timing SEI messages from the source specified in Timecode Config

- **Type:** [H264TimecodeInsertionBehavior](p. 350)
- **Required:** False

**bufSize**

Size of buffer (HRD buffer model) in bits/second.

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

**softness**

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

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

This field indicates how the output video frame rate is specified. If "specified" is selected then the output video frame rate is determined by framerateNumerator and framerateDenominator, else if "initializeFromSource" is selected then the output video frame rate will be set equal to the input video frame rate of the first input.

Type: H264FramerateControl (p. 341)
Required: False

qvbrQualityLevel

Controls the target quality for the video encode. Applies only when the rate control mode is QVBR. Set values for the QVBR quality level field and Max bitrate field that suit your most important viewing devices. Recommended values are: - Primary screen: Quality level: 8 to 10. Max bitrate: 4M - PC or tablet: Quality level: 7. Max bitrate: 1.5M to 3M - Smartphone: Quality level: 6. Max bitrate: 1M to 1.5M

Type: integer
Required: False
Minimum: 1
Maximum: 10

fixedAfd

Four bit AFD value to write on all frames of video in the output stream. Only valid when afdSignaling is set to 'Fixed'.

Type: FixedAfd (p. 338)
Required: False

level

H.264 Level.

Type: H264Level (p. 342)
Required: False

lookAheadRateControl

Amount of lookahead. A value of low can decrease latency and memory usage, while high can produce better quality for certain content.

Type: H264LookAheadRateControl (p. 342)
Required: False

profile

H.264 Profile.

Type: H264Profile (p. 343)
Required: False

framerateNumerator

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

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

**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

**entropyEncoding**

Entropy encoding mode. Use cabac (must be in Main or High profile) or cavlc.

**Type**: H264EntropyEncoding (p. 341)  
**Required**: False

**framerateDenominator**

Framerate denominator.

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

**spatialAq**

If set to enabled, adjust quantization within each frame based on spatial variation of content complexity.

**Type**: H264SpatialAq (p. 349)  
**Required**: False

**adaptiveQuantization**

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

**Type**: H264AdaptiveQuantization (p. 341)  
**Required**: False

**colorMetadata**

Includes colorspace metadata in the output.

**Type**: H264ColorMetadata (p. 341)  
**Required**: False

**gopSize**

GOP size (keyframe interval) in units of either frames or seconds per gopSizeUnits. Must be greater than zero.
**numRefFrames**

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

**gopBReference**

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

- **Type**: H264GopBReference (p. 342)
- **Required**: False

**parControl**

This field indicates how the output pixel aspect ratio is specified. If "specified" is selected then the output video pixel aspect ratio is determined by parNumerator and parDenominator, else if "initializeFromSource" is selected then the output pixel aspect ratio will be set equal to the input video pixel aspect ratio of the first input.

- **Type**: H264ParControl (p. 342)
- **Required**: False

**parDenominator**

Pixel Aspect Ratio denominator.

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

**syntax**

Produces a bitstream compliant with SMPTE RP-2027.

- **Type**: H264Syntax (p. 349)
- **Required**: False

**sceneChangeDetect**

Scene change detection. - On: inserts I-frames when scene change is detected. - Off: does not force an I-frame when scene change is detected.

- **Type**: H264SceneChangeDetect (p. 343)
- **Required**: False
scanType

Sets the scan type of the output to progressive or top-field-first interlaced.

Type: H264ScanType (p. 343)
Required: False

flickerAq

If set to enabled, adjust quantization within each frame to reduce flicker or ‘pop’ on I-frames.

Type: H264FlickerAq (p. 341)
Required: False

gopNumBFrames

Number of B-frames between reference frames.

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

rateControlMode

Rate control mode. QVBR: Quality will match the specified quality level except when it is constrained by the maximum bitrate. Recommended if you or your viewers pay for bandwidth. VBR: Quality and bitrate vary, depending on the video complexity. Recommended instead of QVBR if you want to maintain a specific average bitrate over the duration of the channel. CBR: Quality varies, depending on the video complexity. Recommended only if you distribute your assets to devices that cannot handle variable bitrates.

Type: H264RateControlMode (p. 343)
Required: False

H264SpatialAq

H264 Spatial Aq

DISABLED
ENABLED

H264SubGopLength

H264 Sub Gop Length

DYNAMIC
FIXED

H264Syntax

H264 Syntax

DEFAULT
RP2027

**H264TemporalAq**

H264 Temporal Aq

- DISABLED
- ENABLED

**H264TimecodeInsertionBehavior**

H264 Timecode Insertion Behavior

- DISABLED
- PIC_TIMING_SEI

**HlsAdMarkers**

Hls Ad Markers

- ADOBE
- ELEMENTAL
- ELEMENTAL_SCTE35

**HlsAkamaiHttpTransferMode**

Hls Akamai Http Transfer Mode

- CHUNKED
- NON_CHUNKED

**HlsAkamaiSettings**

Hls Akamai Settings

**salt**

Salt for authenticated Akamai.

Type: string  
Required: False

**httpTransferMode**

Specify whether or not to use chunked transfer encoding to Akamai. User should contact Akamai to enable this feature.

Type: HlsAkamaiHttpTransferMode (p. 350)  
Required: False

**numRetries**

Number of retry attempts that will be made before the Live Event is put into an error state.
Properties

Type: integer
Required: False
Minimum: 0

restartDelay

If a streaming output fails, number of seconds to wait until a restart is initiated. A value of 0 means never restart.

Type: integer
Required: False
Minimum: 0
Maximum: 15

connectionRetryInterval

Number of seconds to wait before retrying connection to the CDN if the connection is lost.

Type: integer
Required: False
Minimum: 0

filecacheDuration

Size in seconds of file cache for streaming outputs.

Type: integer
Required: False
Minimum: 0
Maximum: 600

token

Token parameter for authenticated akamai. If not specified, _gda_ is used.

Type: string
Required: False

HlsBasicPutSettings

Hls Basic Put Settings

numRetries

Number of retry attempts that will be made before the Live Event is put into an error state.

Type: integer
Required: False
Minimum: 0

restartDelay

If a streaming output fails, number of seconds to wait until a restart is initiated. A value of 0 means never restart.
Properties

Type: integer
Required: False
Minimum: 0
Maximum: 15

connectionRetryInterval

Number of seconds to wait before retrying connection to the CDN if the connection is lost.

Type: integer
Required: False
Minimum: 0

filecacheDuration

Size in seconds of file cache for streaming outputs.

Type: integer
Required: False
Minimum: 0
Maximum: 600

HlsCaptionLanguageSetting

Hls Caption Language Setting

INSERT
NONE
OMIT

HlsCdnSettings

Hls Cdn Settings

hlsAkamaiSettings

Type: HlsAkamaiSettings (p. 350)
Required: False

hlsWebdavSettings

Type: HlsWebdavSettings (p. 363)
Required: False

hlsBasicPutSettings

Type: HlsBasicPutSettings (p. 351)
Required: False

hlsMediaStoreSettings

Type: HlsMediaStoreSettings (p. 360)
**HlsClientCache**

Hls Client Cache

- DISABLED
- ENABLED

**HlsCodecSpecification**

Hls Codec Specification

- RFC_4281
- RFC_6381

**HlsDirectoryStructure**

Hls Directory Structure

- SINGLE_DIRECTORY
- SUBDIRECTORY_PER_STREAM

**HlsEncryptionType**

Hls Encryption Type

- AES128
- SAMPLE_AES

**HlsGroupSettings**

Hls Group Settings

- segmentsPerSubdirectory

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

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

- ivInManifest

  For use with encryptionType. The IV (Initialization Vector) is a 128-bit number used in conjunction with the key for encrypting blocks. If set to "include", IV is listed in the manifest, otherwise the IV is not in the manifest.

  - **Type**: HlsIvInManifest (p. 360)
  - **Required**: False
**outputSelection**

MANIFESTSANDSEGMENTS: Generates manifests (master manifest, if applicable, and media manifests) for this output group. SEGMENTSONLY: Does not generate any manifests for this output group.

  - **Type:** HlsOutputSelection (p. 361)
  - **Required:** False

**destination**

A directory or HTTP destination for the HLS segments, manifest files, and encryption keys (if enabled).

  - **Type:** OutputLocationRef (p. 391)
  - **Required:** True

**encryptionType**

Encrypts the segments with the given encryption scheme. Exclude this parameter if no encryption is desired.

  - **Type:** HlsEncryptionType (p. 353)
  - **Required:** False

**indexNSegments**

Applies only if Mode field is LIVE. Specifies the maximum number of segments in the media manifest file. After this maximum, older segments are removed from the media manifest. This number must be less than or equal to the Keep Segments field.

  - **Type:** integer
  - **Required:** False
  - **Minimum:** 3

**constantIv**

For use with encryptionType. This is a 128-bit, 16-byte hex value represented by a 32-character text string. If ivSource is set to "explicit" then this parameter is required and is used as the IV for encryption.

  - **Type:** string
  - **Required:** False
  - **MinLength:** 32
  - **MaxLength:** 32

**timedMetadataId3Frame**

Indicates ID3 frame that has the timecode.

  - **Type:** HlsTimedMetadataId3Frame (p. 363)
  - **Required:** False

**baseUrlManifest**

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
captionLanguageSetting

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.

Type: HlsCaptionLanguageSetting (p. 352)
Required: False

minSegmentLength

When set, minimumSegmentLength 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: False
Minimum: 0

mode

If "vod", all segments are indexed and kept permanently in the destination and manifest. If "live", only the number segments specified in keepSegments and indexNSegments are kept; newer segments replace older segments, which may prevent players from rewinding all the way to the beginning of the event. VOD mode uses HLS EXT-X-PLAYLIST-TYPE of EVENT while the channel is running, converting it to a "VOD" type manifest on completion of the stream.

Type: HlsMode (p. 361)
Required: False

keyProviderSettings

The key provider settings.

Type: KeyProviderSettings (p. 370)
Required: False

manifestCompression

When set to gzip, compresses HLS playlist.

Type: HlsManifestCompression (p. 360)
Required: False

ivSource

For use with encryptionType. The IV (Initialization Vector) is a 128-bit number used in conjunction with the key for encrypting blocks. If this setting is "followsSegmentNumber", it will cause the IV to change every segment (to match the segment number). If this is set to "explicit", you must enter a constant IV value.
**Properties**

- **Type:** HlsIvSource (p. 360)
  - **Required:** False

- **tsFileMode**
  - SEGMENTEDFILES: Emit the program as segments - multiple .ts media files. SINGLEFILE: Applies only if Mode field is VOD. Emit the program as a single .ts media file. The media manifest includes #EXT-X-BYTERANGE tags to index segments for playback. A typical use for this value is when sending the output to AWS Elemental MediaConvert, which can accept only a single media file. Playback while the channel is running is not guaranteed due to HTTP server caching.
  - **Type:** HlsTsFileMode (p. 363)
  - **Required:** False

- **manifestDurationFormat**
  - Indicates whether the output manifest should use floating point or integer values for segment duration.
  - **Type:** HlsManifestDurationFormat (p. 360)
  - **Required:** False

- **keyFormatVersions**
  - Either a single positive integer version value or a slash delimited list of version values (1/2/3).
  - **Type:** string
  - **Required:** False

- **streamInfResolution**
  - Include or exclude RESOLUTION attribute for video in EXT-X-STREAM-INF tag of variant manifest.
  - **Type:** HlsStreamInfResolution (p. 363)
  - **Required:** False

- **timestampDeltaMilliseconds**
  - Provides an extra millisecond delta offset to fine tune the timestamps.
  - **Type:** integer
  - **Required:** False
  - **Minimum:** 0

- **baseUrlContent**
  - 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

- **segmentationMode**
  - useInputSegmentation has been deprecated. The configured segment size is always used.
**captionLanguageMappings**

Mapping of up to 4 caption channels to caption languages. Is only meaningful if captionLanguageSetting is set to "insert".

**Type:** Array of type CaptionLanguageMapping (p. 319)  
**Required:** False

**clientCache**

When set to "disabled", sets the #EXT-X-ALLOW-CACHE:no tag in the manifest, which prevents clients from saving media segments for later replay.

**Type:** HlsClientCache (p. 353)  
**Required:** False

**codecSpecification**

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

**Type:** HlsCodecSpecification (p. 353)  
**Required:** False

**keepSegments**

Applies only if Mode field is LIVE. Specifies the number of media segments (.ts files) to retain in the destination directory.

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

**redundantManifest**

ENABLED: The master manifest (.m3u8 file) for each pipeline includes information about both pipelines: first its own media files, then the media files of the other pipeline. This feature allows playout device that support stale manifest detection to switch from one manifest to the other, when the current manifest seems to be stale. There are still two destinations and two master manifests, but both master manifests reference the media files from both pipelines. DISABLED: The master manifest (.m3u8 file) for each pipeline includes information about its own pipeline only. For an HLS output group with MediaPackage as the destination, the DISABLED behavior is always followed. MediaPackage regenerates the manifests it serves to players so a redundant manifest from MediaLive is irrelevant.

**Type:** HlsRedundantManifest (p. 362)  
**Required:** False

**timedMetadataId3Period**

Timed Metadata interval in seconds.

**Type:** integer
Properties

Required: False
Minimum: 0

**programDateTime**
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 timestampOffset.

*Type: HlsProgramDateTime (p. 362)*
*Required: False*

**directoryStructure**
Place segments in subdirectories.

*Type: HlsDirectoryStructure (p. 353)*
*Required: False*

**keyFormat**
The value specifies how the key is represented in the resource identified by the URI. If parameter is absent, an implicit value of "identity" is used. A reverse DNS string can also be given.

*Type: string*
*Required: False*

**inputLossAction**
Parameter that control output group behavior on input loss.

*Type: InputLossActionForHlsOut (p. 366)*
*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 HlsAdMarkers (p. 350)*
*Required: False*

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

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

**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.
Properties

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

**hlsCdnSettings**

Parameters that control interactions with the CDN.

**Type**: [HlsCdnSettings](p. 352)  
**Required**: False

**iFrameOnlyPlaylists**

DISABLED: Do not create an I-frame-only manifest, but do create the master and media manifests (according to the Output Selection field). STANDARD: Create an I-frame-only manifest for each output that contains video, as well as the other manifests (according to the Output Selection field). The I-frame manifest contains a #EXT-X-I-FRAMES-ONLY tag to indicate it is I-frame only, and one or more #EXT-X-BYTERANGE entries identifying the I-frame position. For example, #EXT-X-BYTERANGE:160364@1461888

**Type**: [IFrameOnlyPlaylistType](p. 364)  
**Required**: False

**HlsInputSettings**

Hls Input Settings

**retries**

The number of consecutive times that attempts to read a manifest or segment must fail before the input is considered unavailable.

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

**bandwidth**

When specified the HLS stream with the m3u8 BANDWIDTH that most closely matches this value will be chosen, otherwise the highest bandwidth stream in the m3u8 will be chosen. The bitrate is specified in bits per second, as in an HLS manifest.

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

**retryInterval**

The number of seconds between retries when an attempt to read a manifest or segment fails.

**Type**: integer  
**Required**: False  
**Minimum**: 0
bufferSegments

When specified, reading of the HLS input will begin this many buffer segments from the end (most recently written segment). When not specified, the HLS input will begin with the first segment specified in the m3u8.

Type: integer
Required: False
Minimum: 0

HlsIvInManifest

Hls Iv In Manifest

EXCLUDE
INCLUDE

HlsIvSource

Hls I v Source

EXPLICIT
FOLLOWS_SEGMENT_NUMBER

HlsManifestCompression

Hls Manifest Compression

GZIP
NONE

HlsManifestDurationFormat

Hls Manifest Duration Format

FLOATING_POINT
INTEGER

HlsMediaStoreSettings

Hls Media Store Settings

mediaStoreStorageClass

When set to temporal, output files are stored in non-persistent memory for faster reading and writing.

Type: HlsMediaStoreStorageClass (p. 361)
Required: False

numRetries

Number of retry attempts that will be made before the Live Event is put into an error state.

Type: integer
Required: False
Minimum: 0

**restartDelay**

If a streaming output fails, number of seconds to wait until a restart is initiated. A value of 0 means never restart.

Type: integer
Required: False
Minimum: 0
Maximum: 15

**connectionRetryInterval**

Number of seconds to wait before retrying connection to the CDN if the connection is lost.

Type: integer
Required: False
Minimum: 0

**filecacheDuration**

Size in seconds of file cache for streaming outputs.

Type: integer
Required: False
Minimum: 0
Maximum: 600

**HlsMediaStoreStorageClass**

Hls Media Store Storage Class

TEMPORAL

**HlsMode**

Hls Mode

LIVE
VOD

**HlsOutputSelection**

Hls Output Selection

MANIFESTS_AND_SEGMENTS
SEGMENTS_ONLY

**HlsOutputSettings**

Hls Output Settings
**segmentModifier**

String concatenated to end of segment filenames.

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

**hlsSettings**

Settings regarding the underlying stream. These settings are different for audio-only outputs.

- **Type:** [HlsSettings](p. 362)
- **Required:** True

**nameModifier**

String concatenated to the end of the destination filename. Accepts "Format Identifiers \":#formatIdentifierParameters.

- **Type:** string
- **Required:** False
- **MinLength:** 1

**HlsProgramDateTime**

Hls Program Date Time

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

**HlsRedundantManifest**

Hls Redundant Manifest

- **DISABLED**
- **ENABLED**

**HlsSegmentationMode**

Hls Segmentation Mode

- **USE_INPUT_SEGMENTATION**
- **USE_SEGMENT_DURATION**

**HlsSettings**

Hls Settings

**standardHlsSettings**

- **Type:** [StandardHlsSettings](p. 399)
- **Required:** False
audioOnlyHlsSettings

Type: AudioOnlyHlsSettings (p. 309)
Required: False

HlsStreamInfResolution

Hls Stream Inf Resolution

EXCLUDE
INCLUDE

HlsTimedMetadataId3Frame

Hls Timed Metadata Id3 Frame

NONE
PRIV
TDRL

HlsTsFileMode

Hls Ts File Mode

SEGMENTED_FILES
SINGLE_FILE

HlsWebdavHttpTransferMode

Hls Webdav Http Transfer Mode

CHUNKED
NON_CHUNKED

HlsWebdavSettings

Hls Webdav Settings

httpTransferMode

Specify whether or not to use chunked transfer encoding to WebDAV.

Type: HlsWebdavHttpTransferMode (p. 363)
Required: False

numRetries

Number of retry attempts that will be made before the Live Event is put into an error state.

Type: integer
Required: False
Properties

Minimum: 0

**restartDelay**

If a streaming output fails, number of seconds to wait until a restart is initiated. A value of 0 means never restart.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 15

**connectionRetryInterval**

Number of seconds to wait before retrying connection to the CDN if the connection is lost.

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

**filecacheDuration**

Size in seconds of file cache for streaming outputs.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 600

**IFrameOnlyPlaylistType**

When set to "standard", an I-Frame only playlist will be written out for each video output in the output group. This I-Frame only playlist will contain byte range offsets pointing to the I-frame(s) in each segment.

- **DISABLED**
- **STANDARD**

**InputAttachment**

**inputId**

The ID of the input

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

**inputAttachmentName**

User-specified name for the attachment. This is required if the user wants to use this input in an input switch action.

- **Type**: string
Required: False

**inputSettings**

Settings of an input (caption selector, etc.)

*Type: InputSettings (p. 368)*
*Required: False*

**InputChannelLevel**

Input Channel Level

**inputChannel**

The index of the input channel used as a source.

*Type: integer*
*Required: True*
*Minimum: 0*
*Maximum: 15*

**gain**

Remixing value. Units are in dB and acceptable values are within the range from -60 (mute) and 6 dB.

*Type: integer*
*Required: True*
*Minimum: -60*
*Maximum: 6*

**InputCodec**

codec in increasing order of complexity

MPEG2
AVC
HEVC

**InputDeblockFilter**

Input Deblock Filter

DISABLED
ENABLED

**InputDenoiseFilter**

Input Denoise Filter

DISABLED
ENABLED

InputFilter

Input Filter

AUTO
DISABLED
FORCED

InputLocation

Input Location

passwordParam

key used to extract the password from EC2 Parameter store

Type: string
Required: False

uri

Uniform Resource Identifier - This should be a path to a file accessible to the Live system (eg. a http:// URI) depending on the output type. For example, a RTMP destination should have a uri similar to:
"rtmp://fmsserver/live".

Type: string
Required: True

username

Username if credentials are required to access a file or publishing point. This can be either a plaintext
username, or a reference to an AWS parameter store name from which the username can be retrieved.
AWS Parameter store format: "ssm://<parameter name>"

Type: string
Required: False

InputLossActionForHlsOut

Input Loss Action For Hls Out

EMIT_OUTPUT
PAUSE_OUTPUT

InputLossActionForMsSmoothOut

Input Loss Action For Ms Smooth Out

EMIT_OUTPUT
PAUSE_OUTPUT
InputLossActionForRtmpOut

Input Loss Action For Rtmp Out

- EMIT_OUTPUT
- PAUSE_OUTPUT

InputLossActionForUdpOut

Input Loss Action For Udp Out

- DROP_PROGRAM
- DROP_TS
- EMIT_PROGRAM

InputLossBehavior

Input Loss Behavior

**inputLossImageType**

Indicates whether to substitute a solid color or a slate into the output after input loss exceeds blackFrameMsec.

- **Type**: InputLossImageType (p. 368)
- **Required**: False

**inputLossImageColor**

When input loss image type is "color" this field specifies the color to use. Value: 6 hex characters representing the values of RGB.

- **Type**: string
- **Required**: False
- **MinLength**: 6
- **MaxLength**: 6

**inputLossImageSlate**

When input loss image type is "slate" these fields specify the parameters for accessing the slate.

- **Type**: InputLocation (p. 366)
- **Required**: False

**repeatFrameMsec**

On input loss, the number of milliseconds to repeat the previous picture before substituting black into the output. A value x, where 0 <= x <= 1,000,000 and a value of 1,000,000 will be interpreted as infinite.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
Maximum: 1000000

**blackFrameMsec**
On input loss, the number of milliseconds to substitute black into the output before switching to the frame specified by inputLossImageType. A value $x$, where $0 \leq x \leq 1,000,000$ and a value of 1,000,000 will be interpreted as infinite.

**Type**: integer  
**Required**: False  
**Minimum**: 0  
**Maximum**: 1000000

**InputLossImageType**
Input Loss Image Type
- COLOR
- SLATE

**InputMaximumBitrate**
Maximum input bitrate in megabits per second. Bitrates up to 50 Mbps are supported currently.
- MAX_10_MBPS
- MAX_20_MBPS
- MAX_50_MBPS

**InputResolution**
Input resolution based on lines of vertical resolution in the input; SD is less than 720 lines, HD is 720 to 1080 lines, UHD is greater than 1080 lines
- SD
- HD
- UHD

**InputSettings**
Live Event input parameters. There can be multiple inputs in a single Live Event.

**sourceEndBehavior**
Loop input if it is a file. This allows a file input to be streamed indefinitely.

**Type**: InputSourceEndBehavior (p. 370)  
**Required**: False

**deblockFilter**
Enable or disable the deblock filter when filtering.

**Type**: InputDeblockFilter (p. 365)
Properties

**Required:** False

**audioSelectors**

Used to select the audio stream to decode for inputs that have multiple available.

- **Type:** Array of type AudioSelector (p. 310)
- **Required:** False

**networkInputSettings**

Input settings.

- **Type:** NetworkInputSettings (p. 388)
- **Required:** False

**inputFilter**

Turns on the filter for this input. MPEG-2 inputs have the deblocking filter enabled by default. 1) auto - filtering will be applied depending on input type/quality 2) disabled - no filtering will be applied to the input 3) forced - filtering will be applied regardless of input type

- **Type:** InputFilter (p. 366)
- **Required:** False

**videoSelector**

Informs which video elementary stream to decode for input types that have multiple available.

- **Type:** VideoSelector (p. 405)
- **Required:** False

**filterStrength**

Adjusts the magnitude of filtering from 1 (minimal) to 5 (strongest).

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 5

**denoiseFilter**

Enable or disable the denoise filter when filtering.

- **Type:** InputDenoiseFilter (p. 365)
- **Required:** False

**captionSelectors**

Used to select the caption input to use for inputs that have multiple available.

- **Type:** Array of type CaptionSelector (p. 320)
- **Required:** False
**InputSourceEndBehavior**

Input Source End Behavior

CONTINUE
LOOP

**InputSpecification**

codec

Input codec

*Type: InputCodec (p. 365)*
*Required: False*

resolution

Input resolution, categorized coarsely

*Type: InputResolution (p. 368)*
*Required: False*

maximumBitrate

Maximum input bitrate, categorized coarsely

*Type: InputMaximumBitrate (p. 368)*
*Required: False*

**InternalServiceError**

message

*Type: string*
*Required: False*

**InvalidRequest**

message

*Type: string*
*Required: False*

**KeyProviderSettings**

Key Provider Settings

**staticKeySettings**

*Type: StaticKeySettings (p. 400)*
Required: False

LimitExceeded

message

Type: string
Required: False

LogLevel

The log level the user wants for their channel.

ERROR
WARNING
INFO
DEBUG
DISABLED

M2tsAbsentInputAudioBehavior

M2ts Absent Input Audio Behavior

DROP
ENCODE_SILENCE

M2tsArib

M2ts Arib

DISABLED
ENABLED

M2tsAribCaptionsPidControl

M2ts Arib Captions Pid Control

AUTO
USE_CONFIGURED

M2tsAudioBufferModel

M2ts Audio Buffer Model

ATSC
DVB

M2tsAudioInterval

M2ts Audio Interval
VIDEO_AND_FIXED_INTERVALS
VIDEO_INTERVAL

**M2tsAudioStreamType**

M2ts Audio Stream Type
- ATSC
- DVB

**M2tsBufferModel**

M2ts Buffer Model
- MULTIPLEX
- NONE

**M2tsCcDescriptor**

M2ts Cc Descriptor
- DISABLED
- ENABLED

**M2tsEbifControl**

M2ts Ebif Control
- NONE
- PASSTHROUGH

**M2tsEbpPlacement**

M2ts Ebp Placement
- VIDEO_AND_AUDIO_PIDS
- VIDEO_PID

**M2tsEsRateInPes**

M2ts Es Rate In Pes
- EXCLUDE
- INCLUDE

**M2tsKlv**

M2ts Klv
- NONE
PASSTHROUGH

**M2tsPcrControl**

M2ts Pcr Control

- CONFIGURED_PCR_PERIOD
- PCR_EVERY_PES_PACKET

**M2tsRateMode**

M2ts Rate Mode

- CBR
- VBR

**M2tsScte35Control**

M2ts Scte35 Control

- NONE
- PASSTHROUGH

**M2tsSegmentationMarkers**

M2ts Segmentation Markers

- EBP
- EBP_LEGACY
- NONE
- PSI_SEGSTART
- RAI_ADAPT
- RAI_SEGSTART

**M2tsSegmentationStyle**

M2ts Segmentation Style

- MAINTAIN_CADENCE
- RESET_CADENCE

**M2tsSettings**

M2ts Settings

**audioStreamType**

When set to atsc, uses stream type = 0x81 for AC3 and stream type = 0x87 for EAC3. When set to dvb, uses stream type = 0x06.

*Type: M2tsAudioStreamType (p. 372)*
Required: False

dcmPId
This field is unused and deprecated.

Type: string
Required: False

dvbTeletextPid
Packet Identifier (PID) for input source DVB Teletext data to this output. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

Type: string
Required: False

aribCaptionsPidControl
If set to auto, pid number used for ARIB Captions will be auto-selected from unused pids. If set to useConfigured, ARIB Captions will be on the configured pid number.

Type: M2tsAribCaptionsPidControl (p. 371)
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.

Type: integer
Required: False
Minimum: 0

rateMode
When vbr, does not insert null packets into transport stream to fill specified bitrate. The bitrate setting acts as the maximum bitrate when vbr is set.

Type: M2tsRateMode (p. 373)
Required: False

segmentationTime
The length in seconds of each segment. Required unless markers is set to None_.

Type: number
Required: False
Minimum: 1.0

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. Can be entered as decimal or hexadecimal values. Each PID specified must be in the range of 32 (or 0x20)..8182 (or 0x1ff6).
Properties

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

**audioFramesPerPes**

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

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

**fragmentTime**

The length in seconds of each fragment. Only used with EBP markers.

**Type**: number
**Required**: False
**Minimum**: 0.0

**ebpLookaheadMs**

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

**ebpAudioInterval**

When videoAndFixedIntervals is selected, 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. Only available when EBP Cablelabs segmentation markers are selected. Partitions 1 and 2 will always follow the video interval.

**Type**: M2tsAudioInterval (p. 371)
**Required**: False

**scte35Pid**

Packet Identifier (PID) of the SCTE-35 stream in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

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

**programNum**

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

**pmtInterval**

The number of milliseconds between instances of this table in the output transport stream. Valid values are 0, 10..1000.

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

**pcrPeriod**

Maximum time in milliseconds between Program Clock Reference (PCRs) inserted into the transport stream.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 500

**segmentationStyle**

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 "resetCadence" 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 $segmentationTime seconds. When a segmentation style of "maintainCadence" 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 $segmentationTime seconds. Note that EBP lookahead is a slight exception to this rule.

- **Type**: M2tsSegmentationStyle (p. 373)
- **Required**: False

**ebif**

If set to passthrough, passes any EBIF data from the input source to this output.

- **Type**: M2tsEbifControl (p. 372)
- **Required**: False

**audioBufferModel**

When set to dvb, uses DVB buffer model for Dolby Digital audio. When set to atsc, the ATSC model is used.

- **Type**: M2tsAudioBufferModel (p. 371)
- **Required**: False
**dvbNitSettings**

Inserts DVB Network Information Table (NIT) at the specified table repetition interval.

*Type:* DvbNitSettings (p. 324)
*Required:* False

**absentInputAudioBehavior**

When set to drop, output audio streams will be removed from the program if the selected input audio stream is removed from the input. This allows the output audio configuration to dynamically change based on input configuration. If this is set to encodeSilence, all output audio streams will output encoded silence when not connected to an active input stream.

*Type:* M2tsAbsentInputAudioBehavior (p. 371)
*Required:* False

**timedMetadataBehavior**

When set to passthrough, timed metadata will be passed through from input to output.

*Type:* M2tsTimedMetadataBehavior (p. 381)
*Required:* False

**timedMetadataPid**

Packet Identifier (PID) of the timed metadata stream in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

*Type:* string
*Required:* False

**pmtPid**

Packet Identifier (PID) for the Program Map Table (PMT) in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

*Type:* string
*Required:* False

**etvSignalPid**

Packet Identifier (PID) for input source ETV Signal data to this output. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

*Type:* string
*Required:* False

**bufferModel**

If set to multiplex, use multiplex buffer model for accurate interleaving. Setting to bufferModel to none can lead to lower latency, but low-memory devices may not be able to play back the stream without interruptions.

*Type:* M2tsBufferModel (p. 372)
*Required:* False
scte35Control
Optionally pass SCTE-35 signals from the input source to this output.

Type: M2tsScte35Control (p. 373)
Required: False

ebpPlacement
Controls placement of EBP on Audio PIDs. If set to videoAndAudioPids, EBP markers will be placed on the video PID and all audio PIDs. If set to videoPid, EBP markers will be placed on only the video PID.

Type: M2tsEbpPlacement (p. 372)
Required: False

arib
When set to enabled, uses ARIB-compliant field muxing and removes video descriptor.

Type: M2tsArib (p. 371)
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
Minimum: 0.0

dvbSdtSettings
Inserts DVB Service Description Table (SDT) at the specified table repetition interval.

Type: DvbSdtSettings (p. 324)
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. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20).8182 (or 0x1ff6).

Type: string
Required: False

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

Type: integer
Required: False
Minimum: 0
Maximum: 65535
pcrControl
When set to pcrEveryPesPacket, 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.

Type: M2tsPcrControl (p. 373)
Required: False

videoPid
Packet Identifier (PID) of the elementary video stream in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

Type: string
Required: False

esRateInPes
Include or exclude the ES Rate field in the PES header.

Type: M2tsEsRateInPes (p. 372)
Required: False

segmentationMarkers
Inserts segmentation markers at each segmentationTime period. raiSegstart sets the Random Access Indicator bit in the adaptation field. raiAdapt sets the RAI bit and adds the current timecode in the private data bytes. psiSegstart 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. ebpLegacy adds Encoder Boundary Point information to the adaptation field using a legacy proprietary format.

Type: M2tsSegmentationMarkers (p. 373)
Required: False

dvbTdtSettings
Inserts DVB Time and Date Table (TDT) at the specified table repetition interval.

Type: DvbTdtSettings (p. 330)
Required: False

klv
If set to passthrough, passes any KLV data from the input source to this output.

Type: M2tsKlv (p. 372)
Required: False

ccDescriptor
When set to enabled, generates captionServiceDescriptor in PMT.

Type: M2tsCcDescriptor (p. 372)
Properties

**patInterval**

The number of milliseconds between instances of this table in the output transport stream. Valid values are 0, 10..1000.

*Type: integer*

*Required: False*

*Minimum: 0*

*Maximum: 1000*

**etvPlatformPid**

Packet Identifier (PID) for input source ETV Platform data to this output. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

*Type: string*

*Required: False*

**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. Can be entered as decimal or hexadecimal values. Each PID specified must be in the range of 32 (or 0x20)..8182 (or 0x1ff6).

*Type: string*

*Required: False*

**aribCaptionsPid**

Packet Identifier (PID) for ARIB Captions in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

*Type: string*

*Required: False*

**scte27Pids**

Packet Identifier (PID) for input source SCTE-27 data to this output. Multiple values are accepted, and can be entered in ranges and/or by comma separation. Can be entered as decimal or hexadecimal values. Each PID specified must be in the range of 32 (or 0x20)..8182 (or 0x1ff6).

*Type: string*

*Required: False*

**klvDataPids**

Packet Identifier (PID) for input source KLV data to this output. Multiple values are accepted, and can be entered in ranges and/or by comma separation. Can be entered as decimal or hexadecimal values. Each PID specified must be in the range of 32 (or 0x20)..8182 (or 0x1ff6).

*Type: string*

*Required: False*
**M2tsTimedMetadataBehavior**

M2ts Timed Metadata Behavior

- NO_PASSTHROUGH
- PASSTHROUGH

**M3u8PcrControl**

M3u8 Pcr Control

- CONFIGURED_PCR_PERIOD
- PCR_EVERY_PES_PACKET

**M3u8Scte35Behavior**

M3u8 Scte35 Behavior

- NO_PASSTHROUGH
- PASSTHROUGH

**M3u8Settings**

Settings information for the .m3u8 container

**pmtPid**

Packet Identifier (PID) for the Program Map Table (PMT) in the transport stream. Can be entered as a decimal or hexadecimal value.

- Type: string
- Required: False

**ecmPid**

This parameter is unused and deprecated.

- Type: string
- Required: False

**scte35Behavior**

If set to passthrough, passes any SCTE-35 signals from the input source to this output.

- Type: [M3u8Scte35Behavior](p. 381)
- 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. Can be entered as a decimal or hexadecimal value.
Properties

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

**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. Can be entered as decimal or hexadecimal values.

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

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

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

**scte35Pid**
Packet Identifier (PID) of the SCTE-35 stream in the transport stream. Can be entered as a decimal or hexadecimal value.

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

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

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

**pcrControl**
When set to pcrEveryPesPacket, 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.

- **Type**: M3u8PcrControl (p. 381)  
  **Required**: False

**videoPid**
Packet Identifier (PID) of the elementary video stream in the transport stream. Can be entered as a decimal or hexadecimal value.

- **Type**: string  
  **Required**: False
pcrPeriod
Maximum time in milliseconds between Program Clock References (PCRs) inserted into the transport stream.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
- **Maximum:** 500

pmtInterval
The number of milliseconds between instances of this table in the output transport stream. A value of "0" writes out the PMT once per segment file.

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

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

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

patInterval
The number of milliseconds between instances of this table in the output transport stream. A value of "0" writes out the PMT once per segment file.

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

timedMetadataPid
Packet Identifier (PID) of the timed metadata stream in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

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

 timedMetadataBehavior
When set to passthrough, timed metadata is passed through from input to output.

- **Type:** M3u8TimedMetadataBehavior (p. 384)
- **Required:** False
M3u8TimedMetadataBehavior

M3u8 Timed Metadata Behavior

- NO_PASSTHROUGH
- PASSTHROUGH

MediaPackageGroupSettings

Media Package Group Settings

destination

MediaPackage channel destination.

- **Type**: OutputLocationRef (p. 391)
- **Required**: True

MediaPackageOutputDestinationSettings

Media Package Output Destination Settings

channelId

ID of the channel in MediaPackage that is the destination for this output group. You do not need to specify the individual inputs in MediaPackage; MediaLive will handle the connection of the two MediaLive pipelines to the two MediaPackage inputs. The MediaPackage channel and MediaLive channel must be in the same region.

- **Type**: string
- **Required**: False
- **MinLength**: 1

MediaPackageOutputSettings

Media Package Output Settings

Mp2CodingMode

Mp2 Coding Mode

- CODING_MODE_1_0
- CODING_MODE_2_0

Mp2Settings

Mp2 Settings

- **codingMode**

The MPEG2 Audio coding mode. Valid values are codingMode10 (for mono) or codingMode20 (for stereo).

- **Type**: Mp2CodingMode (p. 384)
**bitrate**
Average bitrate in bits/second.

- **Type**: number
- **Required**: False

**sampleRate**
Sample rate in Hz.

- **Type**: number
- **Required**: False

**MsSmoothGroupSettings**
Ms Smooth Group Settings

### fragmentLength
Length of mp4 fragments to generate (in seconds). Fragment length must be compatible with GOP size and framerate.

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

**eventId**
MS Smooth event ID to be sent to the IIS server. Should only be specified if eventIdMode is set to useConfigured.

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

**timestampOffset**
Timestamp offset for the event. Only used if timestampOffsetMode is set to useConfiguredOffset.

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

**segmentationMode**
useInputSegmentation has been deprecated. The configured segment size is always used.

- **Type**: SmoothGroupSegmentationMode (p. 399)
- **Required**: False

**numRetries**
Number of retry attempts.
**eventStopBehavior**

When set to sendEos, send EOS signal to IIS server when stopping the event

- **Type**: SmoothGroupEventStopBehavior (p. 399)
- **Required**: False

**acquisitionPointId**

The value of the "Acquisition Point Identity" element used in each message placed in the sparse track. Only enabled if sparseTrackType is not "none".

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

**sparseTrackType**

If set to scte35, use incoming SCTE-35 messages to generate a sparse track in this group of MS-Smooth outputs.

- **Type**: SmoothGroupSparseTrackType (p. 399)
- **Required**: False

**timestampOffsetMode**

Type of timestamp date offset to use. - useEventStartDate: Use the date the event was started as the offset - useConfiguredOffset: Use an explicitly configured date as the offset

- **Type**: SmoothGroupTimestampOffsetMode (p. 399)
- **Required**: False

**destination**

Smooth Streaming publish point on an IIS server. Elemental Live acts as a "Push" encoder to IIS.

- **Type**: OutputLocationRef (p. 391)
- **Required**: True

**audioOnlyTimecodeControl**

If set to passthrough for an audio-only MS Smooth output, the fragment absolute time will be set to the current timecode. This option does not write timecodes to the audio elementary stream.

- **Type**: SmoothGroupAudioOnlyTimecodeControl (p. 398)
- **Required**: False

**connectionRetryInterval**

Number of seconds to wait before retrying connection to the IIS server if the connection is lost. Content will be cached during this time and the cache will be delivered to the IIS server once the connection is re-established.
**Properties**

**filecacheDuration**

Size in seconds of file cache for streaming outputs.

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

**certificateMode**

If set to verifyAuthenticity, verify the https certificate chain to a trusted Certificate Authority (CA). This will cause https outputs to self-signed certificates to fail.

- **Type:** SmoothGroupCertificateMode (p. 398)
- **Required:** False

**inputLossAction**

Parameter that control output group behavior on input loss.

- **Type:** InputLossActionForMsSmoothOut (p. 366)
- **Required:** False

**sendDelayMs**

Number of milliseconds to delay the output from the second pipeline.

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

**eventIdMode**

Specifies whether or not to send an event ID to the IIS server. If no event ID is sent and the same Live Event is used without changing the publishing point, clients might see cached video from the previous run. Options: - "useConfigured" - use the value provided in eventId - "useTimestamp" - generate and send an event ID based on the current timestamp - "noEventId" - do not send an event ID to the IIS server.

- **Type:** SmoothGroupEventIdMode (p. 398)
- **Required:** False

**restartDelay**

Number of seconds before initiating a restart due to output failure, due to exhausting the numRetries on one segment, or exceeding filecacheDuration.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
streamManifestBehavior
When set to send, send stream manifest so publishing point doesn't start until all streams start.

  Type: SmoothGroupStreamManifestBehavior (p. 399)
  Required: False

MsSmoothOutputSettings
Ms Smooth Output Settings

nameModifier
String concatenated to the end of the destination filename. Required for multiple outputs of the same type.

  Type: string
  Required: False

NetworkInputServerValidation
Network Input Server Validation

  CHECK_CRYPTOGRAPHY_AND_VALIDATE_NAME
  CHECK_CRYPTOGRAPHY_ONLY

NetworkInputSettings
Network source to transcode. Must be accessible to the Elemental Live node that is running the live event through a network connection.

hlsInputSettings
Specifies HLS input settings when the uri is for a HLS manifest.

  Type: HlsInputSettings (p. 359)
  Required: False

serverValidation
Check HTTPS server certificates. When set to checkCryptographyOnly, cryptography in the certificate will be checked, but not the server's name. Certain subdomains (notably S3 buckets that use dots in the bucket name) do not strictly match the corresponding certificate's wildcard pattern and would otherwise cause the event to error. This setting is ignored for protocols that do not use https.

  Type: NetworkInputServerValidation (p. 388)
  Required: False

Output
Output settings. There can be multiple outputs within a group.

videoDescriptionName
The name of the VideoDescription used as the source for this output.
outputName
The name used to identify an output.

Type: string
Required: False

MaxLength: 255

captionDescriptionNames
The names of the CaptionDescriptions used as caption sources for this output.

Type: Array of type string
Required: False

audioDescriptionNames
The names of the AudioDescriptions used as audio sources for this output.

Type: Array of type string
Required: False

OutputDestination

mediaPackageSettings
Destination settings for a MediaPackage output; one destination for both encoders.

Type: Array of type MediaPackageOutputDestinationSettings (p. 384)
Required: False

settings
Destination settings for a standard output; one destination for each redundant encoder.

Type: Array of type OutputDestinationSettings (p. 390)
Required: False

id
User-specified id. This is used in an output group or an output.

Type: string
Properties

**Required**: False

**OutputDestinationSettings**

**passwordParam**
key used to extract the password from EC2 Parameter store

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

**streamName**
Stream name for RTMP destinations (URLs of type rtmp://)

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

**url**
A URL specifying a destination

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

**username**
username for destination

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

**OutputGroup**

Output groups for this Live Event. Output groups contain information about where streams should be distributed.

**outputs**

  * **Type**: Array of type Output (p. 388)
  * **Required**: True

**outputGroupSettings**

Settings associated with the output group.

  * **Type**: OutputGroupSettings (p. 391)
  * **Required**: True

**name**
Custom output group name optionally defined by the user. Only letters, numbers, and the underscore character allowed; only 32 characters allowed.
Type: string  
Required: False  
MaxLength: 32

**OutputGroupSettings**

Output Group Settings

**archiveGroupSettings**

Type: ArchiveGroupSettings (p. 304)  
Required: False

**mediaPackageGroupSettings**

Type: MediaPackageGroupSettings (p. 384)  
Required: False

**rtmpGroupSettings**

Type: RtmpGroupSettings (p. 394)  
Required: False

**udpGroupSettings**

Type: UdpGroupSettings (p. 402)  
Required: False

**msSmoothGroupSettings**

Type: MsSmoothGroupSettings (p. 385)  
Required: False

**hlsGroupSettings**

Type: HlsGroupSettings (p. 353)  
Required: False

**frameCaptureGroupSettings**

Type: FrameCaptureGroupSettings (p. 338)  
Required: False

**OutputLocationRef**

Reference to an OutputDestination ID defined in the channel

**destinationRefId**

Type: string  
Required: False
OutputSettings

Output Settings

rtmpOutputSettings

Type: RtmpOutputSettings (p. 395)
Required: False

archiveOutputSettings

Type: ArchiveOutputSettings (p. 304)
Required: False

frameCaptureOutputSettings

Type: FrameCaptureOutputSettings (p. 339)
Required: False

msSmoothOutputSettings

Type: MsSmoothOutputSettings (p. 388)
Required: False

mediaPackageOutputSettings

Type: MediaPackageOutputSettings (p. 384)
Required: False

udpOutputSettings

Type: UdpOutputSettings (p. 402)
Required: False

hlsOutputSettings

Type: HlsOutputSettings (p. 361)
Required: False

PassThroughSettings

Pass Through Settings

RemixSettings

Remix Settings

channelMappings

Mapping of input channels to output channels, with appropriate gain adjustments.

Type: Array of type AudioChannelMapping (p. 305)
Required: True

channelsOut
Number of output channels to be produced. Valid values: 1, 2, 4, 6, 8

  Type: integer
  Required: False
  Minimum: 1
  Maximum: 8

channelsIn
Number of input channels to be used.

  Type: integer
  Required: False
  Minimum: 1
  Maximum: 16

ResourceConflict
message

  Type: string
  Required: False

ResourceNotFound
message

  Type: string
  Required: False

RtmpCacheFullBehavior
Rtmp Cache Full Behavior

  DISCONNECT_IMMEDIATELY
  WAIT_FOR_SERVER

RtmpCaptionData
Rtmp Caption Data

  ALL
  FIELD1_608
  FIELD1_AND_FIELD2_608

RtmpCaptionInfoDestinationSettings
Rtmp Caption Info Destination Settings
**RtmpGroupSettings**

Rtmp Group Settings

**inputLossAction**

Controls the behavior of this RTMP group if input becomes unavailable. - emitOutput: Emit a slate until input returns. - pauseOutput: Stop transmitting data until input returns. This does not close the underlying RTMP connection.

- **Type**: InputLossActionForRtmpOut (p. 367)
- **Required**: False

**captionData**

Controls the types of data that passes to onCaptionInfo outputs. If set to 'all' then 608 and 708 carried DTVCC data will be passed. If set to 'field1AndField2608' then DTVCC data will be stripped out, but 608 data from both fields will be passed. If set to 'field1608' then only the data carried in 608 from field 1 video will be passed.

- **Type**: RtmpCaptionData (p. 393)
- **Required**: False

**authenticationScheme**

Authentication scheme to use when connecting with CDN

- **Type**: AuthenticationScheme (p. 311)
- **Required**: False

**cacheLength**

Cache length, in seconds, is used to calculate buffer size.

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

**restartDelay**

If a streaming output fails, number of seconds to wait until a restart is initiated. A value of 0 means never restart.

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

**cacheFullBehavior**

Controls behavior when content cache fills up. If remote origin server stalls the RTMP connection and does not accept content fast enough the 'Media Cache' will fill up. When the cache reaches the duration specified by cacheLength the cache will stop accepting new content. If set to disconnectImmediately, the RTMP output will force a disconnect. Clear the media cache, and reconnect after restartDelay seconds. If set to waitForServer, the RTMP output will wait up to 5 minutes to allow the origin server to begin accepting data again.
Type: RtmpCacheFullBehavior (p. 393)
Required: False

RtmpOutputCertificateMode

Rtmp Output Certificate Mode

SELF_SIGNED
VERIFY_AUTHENTICITY

RtmpOutputSettings

Rtmp Output Settings

certificateMode

If set to verifyAuthenticity, verify the tls certificate chain to a trusted Certificate Authority (CA). This will cause rtmps outputs with self-signed certificates to fail.

Type: RtmpOutputCertificateMode (p. 395)
Required: False

numRetries

Number of retry attempts.

Type: integer
Required: False
Minimum: 0

destination

The RTMP endpoint excluding the stream name (eg. rtmp://host/appname). For connection to Akamai, a username and password must be supplied. URI fields accept format identifiers.

Type: OutputLocationRef (p. 391)
Required: True

countRetryInterval

Number of seconds to wait before retrying a connection to the Flash Media server if the connection is lost.

Type: integer
Required: False
Minimum: 1

Scte20Convert608To708

Scte20 Convert608 To708

DISABLED
UPCONVERT

**Scte20PlusEmbeddedDestinationSettings**

Scte20 Plus Embedded Destination Settings

**Scte20SourceSettings**

Scte20 Source Settings

**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**

If 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.

Type: Scte20Convert608To708 (p. 395)
Required: False

**Scte27DestinationSettings**

Scte27 Destination Settings

**Scte27SourceSettings**

Scte27 Source Settings

**pid**

The pid field is used in conjunction with the caption selector languageCode field as follows: - Specify PID and Language: Extracts captions from that PID; the language is "informational". - Specify PID and omit Language: Extracts the specified PID. - Omit PID and specify Language: Extracts the specified language, whichever PID that happens to be. - Omit PID and omit Language: Valid only if source is DVB-Sub that is being passed through; all languages will be passed through.

Type: integer
Required: False
Minimum: 1

**Scte35AposNoRegionalBlackoutBehavior**

Scte35 Apos No Regional Blackout Behavior

FOLLOW
**Scte35AposWebDeliveryAllowedBehavior**

Scte35 Apos Web Delivery Allowed Behavior

**adAvailOffset**

When specified, this offset (in milliseconds) is added to the input Ad Avail PTS time. This only applies to embedded SCTE 104/35 messages and does not apply to OOB messages.

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

**webDeliveryAllowedFlag**

When set to ignore, Segment Descriptors with webDeliveryAllowedFlag set to 0 will no longer trigger blackouts or Ad Avail slates

- **Type**: Scte35SpliceInsertWebDeliveryAllowedBehavior (p. 397)
- **Required**: False

**noRegionalBlackoutFlag**

When set to ignore, Segment Descriptors with noRegionalBlackoutFlag set to 0 will no longer trigger blackouts or Ad Avail slates

- **Type**: Scte35SpliceInsertNoRegionalBlackoutBehavior (p. 397)
- **Required**: False

**Scte35SpliceInsertNoRegionalBlackoutBehavior**

Scte35 Splice Insert No Regional Blackout Behavior

**Scte35SpliceInsertWebDeliveryAllowedBehavior**

Scte35 Splice Insert Web Delivery Allowed Behavior

- **Required**: False
Properties

**IGNORE**

**Scte35TimeSignalApos**

Scte35 Time Signal Apos

**adAvailOffset**

When specified, this offset (in milliseconds) is added to the input Ad Avail PTS time. This only applies to embedded SCTE 104/35 messages and does not apply to OOB messages.

*Type: integer*

*Required: False*

*Minimum: -1000*

*Maximum: 1000*

**webDeliveryAllowedFlag**

When set to ignore, Segment Descriptors with webDeliveryAllowedFlag set to 0 will no longer trigger blackouts or Ad Avail slates

*Type: Scte35AposWebDeliveryAllowedBehavior (p. 397)*

*Required: False*

**noRegionalBlackoutFlag**

When set to ignore, Segment Descriptors with noRegionalBlackoutFlag set to 0 will no longer trigger blackouts or Ad Avail slates

*Type: Scte35AposNoRegionalBlackoutBehavior (p. 396)*

*Required: False*

**SmoothGroupAudioOnlyTimecodeControl**

Smooth Group Audio Only Timecode Control

*PASSTHROUGH*

*USE_CONFIGURED_CLOCK*

**SmoothGroupCertificateMode**

Smooth Group Certificate Mode

*SELF_SIGNED*

*VERIFY_AUTHENTICITY*

**SmoothGroupEventIdMode**

Smooth Group Event Id Mode

*NO_EVENT_ID*
Properties

**USE_CONFIGURED**
**USE_TIMESTAMP**

**SmoothGroupEventStopBehavior**
Smooth Group Event Stop Behavior

- NONE
- SEND_EOS

**SmoothGroupSegmentationMode**
Smooth Group Segmentation Mode

- USE_INPUT_SEGMENTATION
- USE_SEGMENT_DURATION

**SmoothGroupSparseTrackType**
Smooth Group Sparse Track Type

- NONE
- SCTE_35

**SmoothGroupStreamManifestBehavior**
Smooth Group Stream Manifest Behavior

- DO_NOT_SEND
- SEND

**SmoothGroupTimestampOffsetMode**
Smooth Group Timestamp Offset Mode

- USE_CONFIGUREDOFFSET
- USE_EVENT_START_DATE

**SmpteTtDestinationSettings**
Smpte Tt Destination Settings

**StandardHlsSettings**
Standard Hls Settings

**m3u8Settings**

Type: M3u8Settings (p. 381)
Required: True
**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

**StaticKeySettings**

Static Key Settings

**staticKeyValue**

Static key value as a 32 character hexadecimal string.

- **Type**: string
- **Required**: True
- **MinLength**: 32
- **MaxLength**: 32

**keyProviderServer**

The URL of the license server used for protecting content.

- **Type**: InputLocation (p. 366)
- **Required**: False

**Tags**

key-value pairs

- **Type**: string

**TeletextDestinationSettings**

Teletext Destination Settings

**TeletextSourceSettings**

Teletext Source Settings

**pageNumber**

Specifies the teletext page number within the data stream from which to extract captions. Range of 0x100 (256) to 0x8FF (2303). Unused for passthrough. Should be specified as a hexadecimal string with no "0x" prefix.

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

**TimecodeConfig**

Timecode Config
**syncThreshold**

Threshold in frames beyond which output timecode is resynchronized to the input timecode. Discrepancies below this threshold are permitted to avoid unnecessary discontinuities in the output timecode. No timecode sync when this is not specified.

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 1000000

**source**

Identifies the source for the timecode that will be associated with the events outputs. - Embedded (embedded): Initialize the output timecode with timecode from the the source. If no embedded timecode is detected in the source, the system falls back to using "Start at 0" (zerobased). - System Clock (systemclock): Use the UTC time. - Start at 0 (zerobased): The time of the first frame of the event will be 00:00:00:00.

- **Type**: TimecodeConfigSource (p. 401)
- **Required**: True

**TimecodeConfigSource**

Timecode Config Source

- EMBEDDED
- SYSTEMCLOCK
- ZEROBASED

**TtmlDestinationSettings**

Ttml Destination Settings

**styleControl**

When set to passthrough, passes through style and position information from a TTML-like input source (TTML, SMPTE-TT, CFF-TT) to the CFF-TT output or TTML output.

- **Type**: TtmlDestinationStyleControl (p. 401)
- **Required**: False

**TtmlDestinationStyleControl**

Ttml Destination Style Control

- PASSTHROUGH
- USE_CONFIGURED

**UdpContainerSettings**

Udp Container Settings
**Properties**

**m2tsSettings**

*Type:* M2tsSettings (p. 373)  
*Required:* False

**UdpGroupSettings**

Udp Group Settings

**inputLossAction**

Specifies behavior of last resort when input video is lost, and no more backup inputs are available. When dropTs is selected the entire transport stream will stop being emitted. When dropProgram is selected the program can be dropped from the transport stream (and replaced with null packets to meet the TS bitrate requirement). Or, when emitProgram is chosen the transport stream will continue to be produced normally with repeat frames, black frames, or slate frames substituted for the absent input video.

*Type:* InputLossActionForUdpOut (p. 367)  
*Required:* False

**timedMetadataId3Frame**

Indicates ID3 frame that has the timecode.

*Type:* UdpTimedMetadataId3Frame (p. 403)  
*Required:* False

**timedMetadataId3Period**

Timed Metadata interval in seconds.

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

**UdpOutputSettings**

Udp Output Settings

**destination**

Destination address and port number for RTP or UDP packets. Can be unicast or multicast RTP or UDP (eg. rtp://239.10.10.10:5001 or udp://10.100.100.100:5002).

*Type:* OutputLocationRef (p. 391)  
*Required:* True

**bufferMsec**

UDP output buffering in milliseconds. Larger values increase latency through the transcoder but simultaneously assist the transcoder in maintaining a constant, low-jitter UDP/RTP output while accommodating clock recovery, input switching, input disruptions, picture reordering, etc.

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

containerSettings
  Type: UdpContainerSettings (p. 401)  
  Required: True

fecOutputSettings
Settings for enabling and adjusting Forward Error Correction on UDP outputs.
  Type: FecOutputSettings (p. 337)  
  Required: False

UdpTimedMetadataId3Frame
Udp Timed Metadata Id3 Frame
  NONE
  PRIV
  TDRL

UpdateChannelClass

destinations
A list of output destinations for this channel.
  Type: Array of type OutputDestination (p. 389)  
  Required: False

channelClass
The channel class that you wish to update this channel to use.
  Type: ChannelClass (p. 323)  
  Required: True

UpdateChannelResultModel
The updated channel's description.

channel
  Type: Channel (p. 321)  
  Required: False

ValidationError

errorMessage
  Type: string
### Required: False

**elementPath**

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

---

**VideoCodecSettings**

Video Codec Settings

**h264Settings**

**Type:** H264Settings (p. 343)  
**Required:** False

**frameCaptureSettings**

**Type:** FrameCaptureSettings (p. 339)  
**Required:** False

---

### VideoDescription

Video settings for this stream.

**respondToAfd**

Indicates how to respond to the AFD values in the input stream. RESPOND causes input video to be clipped, depending on the AFD value, input display aspect ratio, and output display aspect ratio, and (except for FRAMECAPTURE codec) includes the values in the output. PASSTHROUGH (does not apply to FRAMECAPTURE codec) ignores the AFD values and includes the values in the output, so input video is not clipped. NONE ignores the AFD values and does not include the values through to the output, so input video is not clipped.

**Type:** VideoDescriptionRespondToAfd (p. 405)  
**Required:** False

**scalingBehavior**

STRETCHTOOUTPUT configures the output position to stretch the video to the specified output resolution (height and width). This option will override any position value. DEFAULT may insert black boxes (pillar boxes or letter boxes) around the video to provide the specified output resolution.

**Type:** VideoDescriptionScalingBehavior (p. 405)  
**Required:** False

---

**name**

The name of this VideoDescription. Outputs will use this name to uniquely identify this Description. Description names should be unique within this Live Event.

**Type:** string  
**Required:** True
width
Output video width, in pixels. Must be an even number. For most codecs, you can leave this field and height blank in order to use the height and width (resolution) from the source. Note, however, that leaving blank is not recommended. For the Frame Capture codec, height and width are required.

    Type: integer
    Required: False

sharpness
Changes the strength of the anti-alias filter used for scaling. 0 is the softest setting, 100 is the sharpest. A setting of 50 is recommended for most content.

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

codecSettings
Video codec settings.

    Type: VideoCodecSettings (p. 404)
    Required: False

height
Output video height, in pixels. Must be an even number. For most codecs, you can leave this field and width blank in order to use the height and width (resolution) from the source. Note, however, that leaving blank is not recommended. For the Frame Capture codec, height and width are required.

    Type: integer
    Required: False

VideoDescriptionRespondToAfd
Video Description Respond To Afd

    NONE
    PASSTHROUGH
    RESPOND

VideoDescriptionScalingBehavior
Video Description Scaling Behavior

    DEFAULT
    STRETCH_TO_OUTPUT

VideoSelector
Specifies a particular video stream within an input source. An input may have only a single video selector.
**colorSpace**

Specifies the colorspace of an input. This setting works in tandem with colorSpaceConversion to determine if any conversion will be performed.

*Type: VideoSelectorColorSpace (p. 406)*  
*Required: False*

**selectorSettings**

The video selector settings.

*Type: VideoSelectorSettings (p. 407)*  
*Required: False*

**colorSpaceUsage**

Applies only if colorSpace is a value other than follow. This field controls how the value in the colorSpace field will be used. fallback means that when the input does include color space data, that data will be used, but when the input has no color space data, the value in colorSpace will be used. Choose fallback if your input is sometimes missing color space data, but when it does have color space data, that data is correct. force means to always use the value in colorSpace. Choose force if your input usually has no color space data or might have unreliable color space data.

*Type: VideoSelectorColorSpaceUsage (p. 406)*  
*Required: False*

**VideoSelectorColorSpace**

Video Selector Color Space

- FOLLOW
- REC_601
- REC_709

**VideoSelectorColorSpaceUsage**

Video Selector Color Space Usage

- Fallback
- Force

**VideoSelectorPid**

Video Selector Pid

**pid**

Selects a specific PID from within a video source.

*Type: integer*  
*Required: False*
VideoSelectorProgramId

Video Selector Program Id

**programId**

Selects a specific program from within a multi-program transport stream. If the program doesn't exist, the first program within the transport stream will be selected by default.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 65536

VideoSelectorSettings

Video Selector Settings

**videoSelectorPid**

- **Type**: VideoSelectorPid (p. 406)
- **Required**: False

**videoSelectorProgramId**

- **Type**: VideoSelectorProgramId (p. 407)
- **Required**: False

WebvttDestinationSettings

Webvtt Destination Settings

Channels channelId Schedule

**URI**

/prod/channels/{channelId}/schedule

**HTTP Methods**

**GET**

Operation ID: DescribeSchedule

Get a channel schedule
### HTTP Methods

#### Path Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>channelId</td>
<td>String</td>
<td>True</td>
<td>channel ID</td>
</tr>
</tbody>
</table>

#### Query Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>

#### Responses

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>ScheduleDescribeResult (p. 411)</td>
<td>An array of channel schedule actions.</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 415)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 415)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>404</td>
<td>ResourceNotFound (p. 415)</td>
<td>The channel you're requesting to describe does not exist.</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 416)</td>
<td>Request limit exceeded on list channel calls to channel service.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 416)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>BadGatewayException (p. 416)</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>504</td>
<td>GatewayTimeoutException (p. 416)</td>
<td>Gateway Timeout Error</td>
</tr>
</tbody>
</table>

#### PUT

Operation ID: BatchUpdateSchedule

Update a channel schedule

#### Path Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>channelId</td>
<td>String</td>
<td>True</td>
<td>channel ID</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>BatchUpdateScheduleResult (p. 412)</td>
<td>Successful update of the schedule.</td>
</tr>
</tbody>
</table>
### DELETE

**Operation ID:** DeleteSchedule

Delete all schedule actions on a channel.

**Path Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>channelId</td>
<td>String</td>
<td>True</td>
<td>channel ID</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>ScheduleDeleteResultModel</td>
<td>Successful delete of the schedule.</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 415)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 415)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>404</td>
<td>ResourceNotFound (p. 415)</td>
<td>The channel you're requesting to describe does not exist.</td>
</tr>
<tr>
<td>422</td>
<td>ChannelConfigurationValidationError</td>
<td>The Channel failed validation and could not be created.</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 416)</td>
<td>Request limit exceeded on list channel calls to channel service.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 416)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>BadGatewayException (p. 416)</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>504</td>
<td>GatewayTimeoutException</td>
<td>Gateway Timeout Error</td>
</tr>
</tbody>
</table>
Schemas

Request Bodies

Example PUT

```json
{
  "deletes": {
    "actionNames": [
      "string"
    ]
  },
  "creates": {  
    "scheduleActions": [
      {
        "scheduleActionStartSettings": {
          "followModeScheduleActionStartSettings": {
            "referenceActionName": "string",
            "followPoint": enum
          },
          "fixedModeScheduleActionStartSettings": {
            "time": "string"
          }
        },
        "actionName": "string",
        "scheduleActionSettings": {
          "staticImageDeactivateSettings": {
            "fadeOut": integer,
            "layer": integer
          },
          "hlsTimedMetadataSettings": {
            "id3": "string"
          },
          "scte35SpliceInsertSettings": {
            "spliceEventId": integer,
            "duration": integer
          },
          "inputSwitchSettings": {
            "inputAttachmentNameReference": "string"
          },
          "scte35TimeSignalSettings": {
            "scte35Descriptors": [  
              {
                "scte35DescriptorSettings": {
                  "segmentationDescriptorScte35DescriptorSettings": {
                    "subSegmentsExpected": integer,
                    "segmentationEventId": integer,
                    "segmentationDuration": integer,
                    "segmentationCancelIndicator": enum,
                    "subSegmentNum": integer,
                    "segmentationUpidType": integer,
                    "segmentNum": integer,
                    "deliveryRestrictions": {
                      "deviceRestrictions": enum,
                      "webDeliveryAllowedFlag": enum,
                      "noRegionalBlackoutFlag": enum,
                      "archiveAllowedFlag": enum
                    },
                    "segmentationUpid": "string",
                    "segmentationTypeId": integer,
                    "segmentsExpected": integer
                  }
                }
              }
            ]
          }
        }
      }
    ]
  }
}```
"staticImageActivateSettings": {
  "duration": integer,
  "image": {
    "passwordParam": "string",
    "uri": "string",
    "username": "string"
  },
  "fadeOut": integer,
  "imageY": integer,
  "fadeIn": integer,
  "imageX": integer,
  "width": integer,
  "opacity": integer,
  "layer": integer,
  "height": integer
},
"scte35ReturnToNetworkSettings": {
  "spliceEventId": integer
},
"pauseStateSettings": {
  "pipelines": [
    {
      "pipelineId": enum
    }
  ]
}
}

Response Bodies

Example ScheduleDescribeResultModel

{
  "nextToken": "string",
  "scheduleActions": [
    {
      "scheduleActionStartSettings": {
        "followModeScheduleActionStartSettings": {
          "referenceActionName": "string",
          "followPoint": enum
        },
        "fixedModeScheduleActionStartSettings": {
          "time": "string"
        }
      },
      "actionName": "string",
      "scheduleActionSettings": {
        "staticImageDeactivateSettings": {
          "fadeOut": integer,
          "layer": integer
        },
        "hlsTimedMetadataSettings": {
          "id3": "string"
        },
        "scte35SpliceInsertSettings": {
          "spliceEventId": integer,
          "duration": integer
        }
      }
    }
  ]
}
"inputSwitchSettings": {
  "inputAttachmentNameReference": "string"
},
"scte35TimeSignalSettings": {
  "scte35Descriptors": [
    {
      "scte35DescriptorSettings": {
        "segmentationDescriptorScte35DescriptorSettings": {
          "subSegmentsExpected": integer,
          "segmentationEventId": integer,
          "segmentationDuration": integer,
          "segmentationCancelIndicator": enum,
          "subSegmentNum": integer,
          "segmentationUpidType": integer,
          "segmentNum": integer,
          "deliveryRestrictions": {
            "deviceRestrictions": enum,
            "webDeliveryAllowedFlag": enum,
            "noRegionalBlackoutFlag": enum,
            "archiveAllowedFlag": enum
          },
          "segmentationUpid": "string",
          "segmentationTypeId": integer,
          "segmentsExpected": integer
        }
      }
    }
  ],
  "staticImageActivateSettings": {
    "duration": integer,
    "image": {
      "passwordParam": "string",
      "url": "string",
      "username": "string"
    },
    "fadeOut": integer,
    "imageY": integer,
    "fadeIn": integer,
    "imageX": integer,
    "width": integer,
    "opacity": integer,
    "layer": integer,
    "height": integer
  },
  "scte35ReturnToNetworkSettings": {
    "spliceEventId": integer
  },
  "pauseStateSettings": {
    "pipelines": [
    {
      "pipelineId": enum
    }
  ]
}
}

Example BatchUpdateScheduleResult

{
}

412
"deletes": {
  "scheduleActions": [
    {
      "scheduleActionStartSettings": {
        "followModeScheduleActionStartSettings": {
          "referenceActionName": "string",
          "followPoint": enum
        },
        "fixedModeScheduleActionStartSettings": {
          "time": "string"
        }
      },
      "actionName": "string",
      "scheduleActionSettings": {
        "staticImageDeactivateSettings": {
          "fadeOut": integer,
          "layer": integer
        },
        "hlsTimedMetadataSettings": {
          "id3": "string"
        },
        "scte35SpliceInsertSettings": {
          "spliceEventId": integer,
          "duration": integer
        },
        "inputSwitchSettings": {
          "inputAttachmentNameReference": "string"
        },
        "scte35TimeSignalSettings": {
          "scte35Descriptors": [
            {
              "scte35DescriptorSettings": {
                "segmentationDescriptorScte35DescriptorSettings": {
                  "subSegmentsExpected": integer,
                  "segmentationEventId": integer,
                  "segmentationDuration": integer,
                  "segmentationCancelIndicator": enum,
                  "subSegmentNum": integer,
                  "segmentationUpidType": integer,
                  "segmentNum": integer,
                  "deliveryRestrictions": {
                    "deviceRestrictions": enum,
                    "webDeliveryAllowedFlag": enum,
                    "noRegionalBlackoutFlag": enum,
                    "archiveAllowedFlag": enum
                  },
                  "segmentationUpid": "string",
                  "segmentationTypeId": integer,
                  "segmentsExpected": integer
                }
              }
            }
          ]
        }
      }
    }
  ]
}
"opacity": integer,
"layer": integer,
"height": integer
},
"scte35ReturnToNetworkSettings": {
"spliceEventId": integer
},
"pauseStateSettings": {
"pipelines": [
{
"pipelineId": enum
}
]
}
}
}
]
},
"creates": {
"scheduleActions": [
{
"scheduleActionStartSettings": {
"followModeScheduleActionStartSettings": {
"referenceActionName": "string",
"followPoint": enum
},
"fixedModeScheduleActionStartSettings": {
"time": "string"
}
},
"actionName": "string",
"scheduleActionSettings": {
"staticImageDeactivateSettings": {
"fadeOut": integer,
"layer": integer
},
"hlsTimedMetadataSettings": {
"id3": "string"
},
"scte35SpliceInsertSettings": {
"spliceEventId": integer,
"duration": integer
},
"inputSwitchSettings": {
"inputAttachmentNameReference": "string"
},
"scte35TimeSignalSettings": {
"scte35Descriptors": [
{
"scte35DescriptorSettings": {
"segmentationDescriptorScte35DescriptorSettings": {
"subSegmentsExpected": integer,
"segmentationEventId": integer,
"segmentationDuration": integer,
"segmentationCancelIndicator": enum,
"subSegmentNum": integer,
"segmentUpidType": integer,
"segmentNum": integer,
"deliveryRestrictions": {
"deviceRestrictions": enum,
"webDeliveryAllowedFlag": enum,
"noRegionalBlackoutFlag": enum,
"archiveAllowedFlag": enum
}.
"segmentationUpid": "string",
"segmentationTypeId": integer,
"segmentsExpected": integer
}

"staticImageActivateSettings": {
  "duration": integer,
  "image": {
    "passwordParam": "string",
    "uri": "string",
    "username": "string"
  },
  "fadeOut": integer,
  "imageY": integer,
  "fadeIn": integer,
  "imageX": integer,
  "width": integer,
  "opacity": integer,
  "layer": integer,
  "height": integer
},
"scte35ReturnToNetworkSettings": {
  "spliceEventId": integer
},
"pauseStateSettings": {
  "pipelines": [
    {
      "pipelineId": enum
    }
  ]
}
}
}

Example ScheduleDeleteResultModel

{
  
}

Example InvalidRequest

{
  "message": "string"
}

Example AccessDenied

{
  "message": "string"
}

Example ResourceNotFound

{
  "message": "string"
}
Example ChannelConfigurationValidationError

```json
{
   "validationErrors": [
       {
           "errorMessage": "string",
           "elementPath": "string"
       }
   ],
   "message": "string"
}
```

Example LimitExceeded

```json
{
   "message": "string"
}
```

Example InternalServiceError

```json
{
   "message": "string"
}
```

Example BadGatewayException

```json
{
   "message": "string"
}
```

Example GatewayTimeoutException

```json
{
   "message": "string"
}
```

Properties

AccessDenied

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

BadGatewayException

- **message**
  - Type: string
  - Required: False
BatchScheduleActionCreateRequest
A list of schedule actions to create (in a request) or that have been created (in a response).

scheduleActions
A list of schedule actions to create.

Type: Array of type ScheduleAction (p. 421)
Required: True

BatchScheduleActionCreateResult
List of actions that have been created in the schedule.

scheduleActions
List of actions that have been created in the schedule.

Type: Array of type ScheduleAction (p. 421)
Required: True

BatchScheduleActionDeleteRequest
A list of schedule actions to delete.

actionNames
A list of schedule actions to delete.

Type: Array of type string
Required: True

BatchScheduleActionDeleteResult
List of actions that have been deleted from the schedule.

scheduleActions
List of actions that have been deleted from the schedule.

Type: Array of type ScheduleAction (p. 421)
Required: True

BatchUpdateScheduleRequest
A request to create actions (add actions to the schedule), delete actions (remove actions from the schedule), or both create and delete actions.

deletes
Schedule actions to delete from the schedule.

Type: BatchScheduleActionDeleteRequest (p. 417)
Required: False

creates
Schedule actions to create in the schedule.

Type: BatchScheduleActionCreateRequest (p. 417)
Required: False

BatchUpdateScheduleResult
Results of a batch schedule update.

deletes
Schedule actions deleted from the schedule.

Type: BatchScheduleActionDeleteResult (p. 417)
Required: False

creates
Schedule actions created in the schedule.

Type: BatchScheduleActionCreateResult (p. 417)
Required: False

ChannelConfigurationValidationError
validationErrors
A collection of validation error responses.

Type: Array of type ValidationError (p. 431)
Required: False

message

Type: string
Required: False

FixedModeScheduleActionStartSettings
Start time for the action.

time
Start time for the action to start in the channel. (Not the time for the action to be added to the schedule: actions are always added to the schedule immediately.) UTC format: yyyy-mm-ddThh:mm:ss.nnnZ. All the letters are digits (for example, mm might be 01) except for the two constants "T" for time and "Z" for "UTC format".

Type: string
**FollowModeScheduleActionStartSettings**

Settings to specify if an action follows another.

**referenceActionName**

The action name of another action that this one refers to.

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

**followPoint**

Identifies whether this action starts relative to the start or relative to the end of the reference action.

- **Type**: `FollowPoint (p. 419)`
- **Required**: True

**GatewayTimeoutException**

**message**

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

**HlsTimedMetadataScheduleActionSettings**

Settings for the action to emit HLS metadata

**id3**

Base64 string formatted according to the ID3 specification: http://id3.org/id3v2.4.0-structure

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

**InputLocation**

Input Location

**passwordParam**

key used to extract the password from EC2 Parameter store
**Properties**

**uri**

Uniform Resource Identifier - This should be a path to a file accessible to the Live system (eg. a http:// URI) depending on the output type. For example, a RTMP destination should have a uri similar to: "rtmp://fmsserver/live".

**username**

Username if credentials are required to access a file or publishing point. This can be either a plaintext username, or a reference to an AWS parameter store name from which the username can be retrieved. AWS Parameter store format: "ssm://<parameter name>"

**InputSwitchScheduleActionSettings**

Settings for the action to switch an input.

**inputAttachmentNameReference**

The name of the input attachment that should be switched to by this action.

**InternalServiceError**

**message**

**InvalidRequest**

**message**

**LimitExceeded**

**message**
PauseStateScheduleActionSettings

Settings for the action to set pause state of a channel.

**pipelines**

*Type:* Array of type PipelinePauseStateSettings (p. 421)

*Required:* False

PipelineId

Pipeline ID

*PIPELINE_0*

*PIPELINE_1*

PipelinePauseStateSettings

Settings for pausing a pipeline.

**pipelineId**

Pipeline ID to pause ("PIPELINE_0" or "PIPELINE_1").

*Type:* PipelineId (p. 421)

*Required:* True

ResourceNotFound

**message**

*Type:* string

*Required:* False

ScheduleAction

Contains information on a single schedule action.

**scheduleActionStartSettings**

The time for the action to start in the channel.

*Type:* ScheduleActionStartSettings (p. 423)

*Required:* True

**actionName**

The name of the action, must be unique within the schedule. This name provides the main reference to an action once it is added to the schedule. A name is unique if it is no longer in the schedule. The schedule is automatically cleaned up to remove actions with a start time of more than 1 hour ago (approximately) so at that point a name can be reused.

*Type:* string
Required: True

scheduleActionSettings
Settings for this schedule action.
Type: ScheduleActionSettings (p. 422)
Required: True

ScheduleActionSettings
Holds the settings for a single schedule action.

staticImageDeactivateSettings
Action to deactivate a static image overlay
Type: StaticImageDeactivateScheduleActionSettings (p. 430)
Required: False

hlsTimedMetadataSettings
Action to insert HLS metadata
Type: HlsTimedMetadataScheduleActionSettings (p. 419)
Required: False

scte35SpliceInsertSettings
Action to insert SCTE-35 splice_insert message
Type: Scte35SpliceInsertScheduleActionSettings (p. 428)
Required: False

inputSwitchSettings
Action to switch the input
Type: InputSwitchScheduleActionSettings (p. 420)
Required: False

scte35TimeSignalSettings
Action to insert SCTE-35 time_signal message
Type: Scte35TimeSignalScheduleActionSettings (p. 428)
Required: False

staticImageActivateSettings
Action to activate a static image overlay
Type: StaticImageActivateScheduleActionSettings (p. 429)
Properties

**Required**: False

**scte35ReturnToNetworkSettings**
Action to insert SCTE-35 return_to_network message

*Type*: `Scte35ReturnToNetworkScheduleActionSettings (p. 425)`
*Required*: False

**pauseStateSettings**
Action to pause or unpause one or both channel pipelines

*Type*: `PauseStateScheduleActionSettings (p. 421)`
*Required*: False

**ScheduleActionStartSettings**
Settings to specify the start time for an action.

**followModeScheduleActionStartSettings**
Specifies an action to follow for scheduling this action.

*Type*: `FollowModeScheduleActionStartSettings (p. 419)`
*Required*: False

**fixedModeScheduleActionStartSettings**
Holds the start time for the action.

*Type*: `FixedModeScheduleActionStartSettings (p. 418)`
*Required*: False

**ScheduleDeleteResultModel**
Result of a schedule deletion.

**ScheduleDescribeResultModel**
Results of a schedule describe.

**nextToken**
The next token; for use in pagination.

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

**scheduleActions**
The list of actions in the schedule.

*Type*: Array of type `ScheduleAction (p. 421)`
**Properties**

**Required:** True

### Scte35ArchiveAllowedFlag

Corresponds to the `archive_allowed` parameter. A value of `ARCHIVE_NOT_ALLOWED` corresponds to 0 (false) in the SCTE-35 specification. If you include one of the "restriction" flags then you must include all four of them.

- `ARCHIVE_NOT_ALLOWED`
- `ARCHIVE_ALLOWED`

### Scte35DeliveryRestrictions

Corresponds to SCTE-35 `delivery_not_restricted_flag` parameter. To declare delivery restrictions, include this element and its four "restriction" flags. To declare that there are no restrictions, omit this element.

#### deviceRestrictions

Corresponds to SCTE-35 `device_restrictions` parameter.

- **Type:** `Scte35DeviceRestrictions` (p. 425)
- **Required:** True

### webDeliveryAllowedFlag

Corresponds to SCTE-35 `web_delivery_allowed_flag` parameter.

- **Type:** `Scte35WebDeliveryAllowedFlag` (p. 428)
- **Required:** True

### noRegionalBlackoutFlag

Corresponds to SCTE-35 `no_regional_blackout_flag` parameter.

- **Type:** `Scte35NoRegionalBlackoutFlag` (p. 425)
- **Required:** True

### archiveAllowedFlag

Corresponds to SCTE-35 `archive_allowed_flag`.

- **Type:** `Scte35ArchiveAllowedFlag` (p. 424)
- **Required:** True

### Scte35Descriptor

Holds one set of SCTE-35 Descriptor Settings.

#### scte35DescriptorSettings

SCTE-35 Descriptor Settings.

- **Type:** `Scte35DescriptorSettings` (p. 425)

---

424
Required: True

**Scte35DescriptorSettings**

SCTE-35 Descriptor settings.

**segmentationDescriptorScte35DescriptorSettings**

SCTE-35 Segmentation Descriptor.

- **Type**: Scte35SegmentationDescriptor (p. 426)
- **Required**: True

**Scte35DeviceRestrictions**

Corresponds to the device_restrictions parameter in a segmentation_descriptor. If you include one of the "restriction" flags then you must include all four of them.

- NONE
- RESTRICT_GROUP0
- RESTRICT_GROUP1
- RESTRICT_GROUP2

**Scte35NoRegionalBlackoutFlag**

Corresponds to the no_regional_blackout_flag parameter. A value of REGIONAL_BLACKOUT corresponds to 0 (false) in the SCTE-35 specification. If you include one of the "restriction" flags then you must include all four of them.

- REGIONAL_BLACKOUT
- NO_REGIONAL_BLACKOUT

**Scte35ReturnToNetworkScheduleActionSettings**

Settings for a SCTE-35 return_to_network message.

**spliceEventId**

The splice_event_id for the SCTE-35 splice_insert, as defined in SCTE-35.

- **Type**: integer
- **Required**: True
- **Format**: int64
- **Minimum**: 0
- **Maximum**: 4294967295

**Scte35SegmentationCancelIndicator**

Corresponds to SCTE-35 segmentation_event_cancel_indicator. SEGMENTATION_EVENT_NOT_CANCELED corresponds to 0 in the SCTE-35 specification and indicates that this is an insertion request. SEGMENTATION_EVENT_CANCELED corresponds to 1 in the SCTE-35 specification and indicates that this is a cancelation request, in which case complete this field and the existing event ID to cancel.
SEGMENTATION_EVENT_NOT_CANCELED
SEGMENTATION_EVENT_CANCELED

Scte35SegmentationDescriptor

Corresponds to SCTE-35 segmentation_descriptor.

subSegmentsExpected

Corresponds to SCTE-35 sub_segments_expected. A value that is valid for the specified segmentation_type_id.

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

segmentationEventId

Corresponds to SCTE-35 segmentation_event_id.

  Type: integer
  Required: True
  Format: int64
  Minimum: 0
  Maximum: 4294967295

segmentationDuration

Corresponds to SCTE-35 segmentation_duration. Optional. The duration for the time_signal, in 90 KHz ticks. To convert seconds to ticks, multiple the seconds by 90,000. Enter time in 90 KHz clock ticks. If you do not enter a duration, the time_signal will continue until you insert a cancellation message.

  Type: integer
  Required: False
  Format: int64
  Minimum: 0
  Maximum: 1099511627775

segmentationCancelIndicator

Corresponds to SCTE-35 segmentation_event_cancel_indicator.

  Type: Scte35SegmentationCancelIndicator (p. 425)
  Required: True

subSegmentNum

Corresponds to SCTE-35 sub_segment_num. A value that is valid for the specified segmentation_type_id.

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

**segmentationUpid**

Corresponds to SCTE-35 segmentation_upid. On the console, enter one of the types listed in the SCTE-35 specification, converted to a decimal. For example, "0x0C" hex from the specification is "12" in decimal. In the CLI, API, or an SDK, enter one of the types listed in the SCTE-35 specification, in either hex (for example, "0x0C") or in decimal (for example, "12").

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

**segmentNum**

Corresponds to SCTE-35 segment_num. A value that is valid for the specified segmentation_type_id.

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

**deliveryRestrictions**

Holds the four SCTE-35 delivery restriction parameters.

- **Type:** Scte35DeliveryRestrictions (p. 424)
- **Required:** False

**segmentationUpid**

Corresponds to SCTE-35 segmentation_upid. Enter a string containing the hexadecimal representation of the characters that make up the SCTE-35 segmentation_upid value. Must contain an even number of hex characters. Do not include spaces between each hex pair. For example, the ASCII "ADS Information" becomes hex "41445320496e666f726d6174696f6e.

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

**segmentationTypeld**

Corresponds to SCTE-35 segmentation_type_id. One of the segmentation_type_id values listed in the SCTE-35 specification. On the console, enter the ID in decimal (for example, "52"). In the CLI, API, or an SDK, enter the ID in hex (for example, "0x34") or decimal (for example, "52").

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

**segmentsExpected**

Corresponds to SCTE-35 segments_expected. A value that is valid for the specified segmentation_type_id.
**Scte35SpliceInsertScheduleActionSettings**

Settings for a SCTE-35 splice_insert message.

**spliceEventId**

The splice_event_id for the SCTE-35 splice_insert, as defined in SCTE-35.

- **Type**: integer
- **Required**: True
- **Format**: int64
- **Minimum**: 0
- **Maximum**: 4294967295

**duration**

Optional, the duration for the splice_insert, in 90 KHz ticks. To convert seconds to ticks, multiply the seconds by 90,000. If you enter a duration, there is an expectation that the downstream system can read the duration and cue in at that time. If you do not enter a duration, the splice_insert will continue indefinitely and there is an expectation that you will enter a return_to_network to end the splice_insert at the appropriate time.

- **Type**: integer
- **Required**: False
- **Format**: int64
- **Minimum**: 0
- **Maximum**: 8589934591

**Scte35TimeSignalScheduleActionSettings**

Settings for a SCTE-35 time_signal.

**scte35Descriptors**

The list of SCTE-35 descriptors accompanying the SCTE-35 time_signal.

- **Type**: Array of type **Scte35Descriptor** (p. 424)
- **Required**: True

**Scte35WebDeliveryAllowedFlag**

Corresponds to the web_delivery_allowed_flag parameter. A value of WEB_DELIVERY_NOT_ALLOWED corresponds to 0 (false) in the SCTE-35 specification. If you include one of the “restriction” flags then you must include all four of them.

- WEB_DELIVERY_NOT_ALLOWED
- WEB_DELIVERY_ALLOWED
**StaticImageActivateScheduleActionSettings**

Settings for the action to activate a static image.

**duration**

The duration in milliseconds for the image to remain on the video. If omitted or set to 0 the duration is unlimited and the image will remain until it is explicitly deactivated.

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

**image**

The location and filename of the image file to overlay on the video. The file must be a 32-bit BMP, PNG, or TGA file, and must not be larger (in pixels) than the input video.

- **Type:** InputLocation (p. 419)
- **Required:** True

**fadeOut**

Applies only if a duration is specified. The time in milliseconds for the image to fade out. The fade-out starts when the duration time is hit, so it effectively extends the duration. Default is 0 (no fade-out).

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

**imageY**

Placement of the top edge of the overlay relative to the top edge of the video frame, in pixels. 0 (the default) is the top edge of the frame. If the placement causes the overlay to extend beyond the bottom edge of the underlying video, then the overlay is cropped on the bottom.

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

**fadeIn**

The time in milliseconds for the image to fade in. The fade-in starts at the start time of the overlay. Default is 0 (no fade-in).

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

**imageX**

Placement of the left edge of the overlay relative to the left edge of the video frame, in pixels. 0 (the default) is the left edge of the frame. If the placement causes the overlay to extend beyond the right edge of the underlying video, then the overlay is cropped on the right.
Properties

**width**

The width of the image when inserted into the video, in pixels. The overlay will be scaled up or down to the specified width. Leave blank to use the native width of the overlay.

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

**opacity**

Opacity of image where 0 is transparent and 100 is fully opaque. Default is 100.

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

**layer**

The number of the layer, 0 to 7. There are 8 layers that can be overlaid on the video, each layer with a different image. The layers are in Z order, which means that overlays with higher values of layer are inserted on top of overlays with lower values of layer. Default is 0.

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

**height**

The height of the image when inserted into the video, in pixels. The overlay will be scaled up or down to the specified height. Leave blank to use the native height of the overlay.

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

**StaticImageDeactivateScheduleActionSettings**

Settings for the action to deactivate the image in a specific layer.

**fadeOut**

The time in milliseconds for the image to fade out. Default is 0 (no fade-out).

- **Type:** integer
- **Required:** False
- **Minimum:** 0
Channels channelId Start

URI

/prod/channels/{channelId}/start

HTTP Methods

POST

Operation ID: StartChannel

Starts an existing channel

Path Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>channelId</td>
<td>String</td>
<td>True</td>
<td>channel ID</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>Channel (p. 432)</td>
<td>Successfully initiated start of the channel.</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 443)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 443)</td>
<td>You do not have permission to list channels.</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>404</td>
<td>ResourceNotFound (p. 443)</td>
<td>The channel you're requesting to describe does not exist.</td>
</tr>
<tr>
<td>409</td>
<td>ResourceConflict (p. 444)</td>
<td>The channel is unable to create due to an issue with channel resources.</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 444)</td>
<td>Request limit exceeded on list channel calls to channel service.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 444)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>BadGatewayException (p. 444)</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>504</td>
<td>GatewayTimeoutException (p. 444)</td>
<td>Gateway Timeout Error</td>
</tr>
</tbody>
</table>

**Schemas**

**Response Bodies**

**Example Channel**

```json
{
  "inputAttachments": [
    {
      "inputId": "string",
      "inputAttachmentName": "string",
      "inputSettings": {
        "sourceEndBehavior": enum,
        "deblockFilter": enum,
        "audioSelectors": [
          {
            "name": "string",
            "selectorSettings": {
              "audioLanguageSelection": {
                "languageSelectionPolicy": enum,
                "languageCode": "string"},
              "audioPidSelection": {
                "pid": integer
              }
            }
          }
        ],
        "networkInputSettings": {
          "hlsInputSettings": {
            "retries": integer,
            "bandwidth": integer,
            "retryInterval": integer,
            "bufferSegments": integer
          },
          "serverValidation": enum
        },
        "inputFilter": enum,
        "videoSelector": {
          "colorSpace": enum,
          "selectorSettings": {
```
"videoSelectorPid": {
  "pid": integer
},
"videoSelectorProgramId": {
  "programId": integer
},
"colorSpaceUsage": enum,
"filterStrength": integer,
"denoiseFilter": enum,
"captionSelectors": [
  {
    "name": "string",
    "languageCode": "string",
    "selectorSettings": {
      "embeddedSourceSettings": {
        "scte20Detection": enum,
        "source608ChannelNumber": integer,
        "convert608To708": enum,
        "source608TrackNumber": integer
      },
      "scte20SourceSettings": {
        "source608ChannelNumber": integer,
        "convert608To708": enum
      },
      "dvbsubSourceSettings": {
        "pid": integer
      },
      "teletextSourceSettings": {
        "pageNumber": "string"
      },
      "aribSourceSettings": {
      },
      "scte27SourceSettings": {
        "pid": integer
      }
    }
  }
],
"destinations": [
  {
    "mediaPackageSettings": [
      {
        "channelId": "string"
      }
    ],
    "settings": [
      {
        "passwordParam": "string",
        "streamName": "string",
        "url": "string",
        "username": "string"
      }
    ],
    "id": "string"
  }
],
"encoderSettings": {
  "timecodeConfig": {
    "syncThreshold": integer,
    "source": enum
  }
}
"outputGroups": [
  {
    "outputs": [
      {
        "videoDescriptionName": "string",
        "outputName": "string",
        "captionDescriptionNames": [
          "string"
        ],
        "outputSettings": {
          "rtmpOutputSettings": {
            "certificateMode": enum,
            "numRetries": integer,
            "destination": {
              "destinationRefId": "string"
            },
            "connectionRetryInterval": integer
          },
          "archiveOutputSettings": {
            "extension": "string",
            "containerSettings": {
              "m2tsSettings": {
                "audioStreamType": enum,
                "ecmPid": "string",
                "dvbTeletextPid": "string",
                "aribCaptionsPidControl": enum,
                "bitrate": integer,
                "rateMode": enum,
                "segmentationTime": number,
                "audioPids": "string",
                "audioFramesPerPes": integer,
                "fragmentTime": number,
                "ebpLookaheadMs": integer,
                "ebpAudioInterval": enum,
                "scte35Pid": "string",
                "programNum": integer,
                "pmtInterval": integer,
                "pcrPeriod": integer,
                "segmentationStyle": enum,
                "ebif": enum,
                "audioBufferModel": enum,
                "dvbNitSettings": {
                  "networkName": "string",
                  "networkId": integer,
                  "repInterval": integer
                },
                "absentInputAudioBehavior": enum,
                "timedMetadataBehavior": enum,
                "pmtPid": "string",
                "etvSignalPid": "string",
                "bufferModel": enum,
                "scte35Control": enum,
                "ebpPlacement": enum,
                "arib": enum,
                "nullPacketBitrate": number,
                "dvbSdtSettings": {
                  "serviceName": "string",
                  "serviceProviderName": "string",
                  "repInterval": integer,
                  "outputSdt": enum
                },
                "pcrPid": "string",
                "transportStreamId": integer,
                "pcrControl": enum,
                "videoPid": "string"
              }
            }
          }
        }
      }
    ]
  }
]
"esRateInPes": enum,
"segmentationMarkers": enum,
"dvbTdtSettings": {
  "repInterval": integer
},
"klv": enum,
"ccDescriptor": enum,
"patInterval": integer,
"etvPlatformPid": "string",
"dvbSubPids": "string",
"aribCaptionsPid": "string",
"scte27Pids": "string",
"klvDataPids": "string"
}
},
"nameModifier": "string"
},
"frameCaptureOutputSettings": {
  "nameModifier": "string"
},
"msSmoothOutputSettings": {
  "nameModifier": "string"
},
"mediaPackageOutputSettings": {
},
"udpOutputSettings": {
  "destination": {
    "destinationRefId": "string"
  },
  "bufferMsec": integer,
  "containerSettings": {
    "m2tsSettings": {
      "audioStreamType": enum,
      "ecmPid": "string",
      "dvbTeletextPid": "string",
      "aribCaptionsPidControl": enum,
      "bitrate": integer,
      "rateMode": enum,
      "segmentationTime": number,
      "audioPids": "string",
      "audioFramesPerPes": integer,
      "fragmentTime": number,
      "ebpLookaheadMs": integer,
      "ebpAudioInterval": enum,
      "scte35Pid": "string",
      "programNum": integer,
      "pmtInterval": integer,
      "pcrPeriod": integer,
      "segmentationStyle": enum,
      "ebif": enum,
      "audioBufferModel": enum,
      "dvbNitSettings": {
        "networkName": "string",
        "networkId": integer,
        "repInterval": integer
      },
      "absentInputAudioBehavior": enum,
      "timedMetadataBehavior": enum,
      "timedMetadataPid": "string",
      "pmtPid": "string",
      "etvSignalPid": "string",
      "bufferModel": enum,
      "scte35Control": enum,
      "ebpPlacement": enum,
      "arib": enum,
      "nullPacketBitrate": number,
"dvbSdtSettings": {  
  "serviceName": "string",
  "serviceProviderName": "string",
  "repInterval": integer,
  "outputSdt": enum
},

"pcrPid": "string",
"transportStreamId": integer,
"pcrControl": enum,
"videoPid": "string",
"esRateInPes": enum,
"segmentationMarkers": enum,
"dvbTdtSettings": {  
  "repInterval": integer
},

"klv": enum,
"ccDescriptor": enum,
"patInterval": integer,
"etvPlatformPid": "string",
"dvbSubPids": "string",
"aribCaptionsPid": "string",
"scte27Pids": "string",
"klvDataPids": "string"
}
",

"fecOutputSettings": {  
  "rowLength": integer,
  "columnDepth": integer,
  "includeFec": enum
}
",

"hlsOutputSettings": {  
  "segmentModifier": "string",
  "hlsSettings": {  
    "standardHlsSettings": {  
      "m3u8Settings": {  
        "pmtPid": "string",
        "ecmPid": "string",
        "scte35Behavior": enum,
        "pcrPid": "string",
        "audioPids": "string",
        "audioFramesPerPes": integer,
        "scte35Pid": "string",
        "transportStreamId": integer,
        "pcrControl": enum,
        "videoPid": "string",
        "pcrPeriod": integer,
        "pmtInterval": integer,
        "programNum": integer,
        "patInterval": integer,
        "timedMetadataPid": "string",
        "timedMetadataBehavior": enum
      },
      "audioRenditionSets": "string"
    },
    "audioOnlyHlsSettings": {  
      "audioTrackType": enum,
      "audioGroupId": "string",
      "audioOnlyImage": {  
        "passwordParam": "string",
        "uri": "string",
        "username": "string"
      }
    }
  }
},

"nameModifier": "string"
"audioDescriptionNames": ["string"],
"outputGroupSettings": {
  "archiveGroupSettings": {
    "destination": {
      "destinationRefId": "string"
    },
    "rolloverInterval": integer
  },
  "mediaPackageGroupSettings": {
    "destination": {
      "destinationRefId": "string"
    }
  },
  "rtmpGroupSettings": {
    "inputLossAction": enum,
    "captionData": enum,
    "authenticationScheme": enum,
    "cacheLength": integer,
    "restartDelay": integer,
    "cacheFullBehavior": enum
  },
  "udpGroupSettings": {
    "inputLossAction": enum,
    "timedMetadataId3Frame": enum,
    "timedMetadataId3Period": integer
  },
  "msSmoothGroupSettings": {
    "fragmentLength": integer,
    "eventId": "string",
    "timestampOffset": "string",
    "segmentationMode": enum,
    "numRetries": integer,
    "eventStopBehavior": enum,
    "acquisitionPointId": "string",
    "sparseTrackType": enum,
    "timestampOffsetMode": enum,
    "destination": {
      "destinationRefId": "string"
    },
    "audioOnlyTimecodeControl": enum,
    "connectionRetryInterval": integer,
    "filecacheDuration": integer,
    "certificateMode": enum,
    "inputLossAction": enum,
    "sendDelayMs": integer,
    "eventIdMode": enum,
    "restartDelay": integer,
    "streamManifestBehavior": enum
  },
  "hlsGroupSettings": {
    "segmentsPerSubdirectory": integer,
    "liviInManifest": enum,
    "outputSelection": enum,
    "destination": {
      "destinationRefId": "string"
    },
    "encryptionType": enum,
    "indexNSegments": integer,
    "constantIV": "string",
    "timedMetadataId3Frame": enum,
"baseUrlManifest": "string",
"captionLanguageSetting": enum,
"minSegmentLength": integer,
"mode": enum,
"keyProviderSettings": {
  "staticKeySettings": {
    "staticKeyValue": "string",
    "keyProviderServer": {
      "passwordParam": "string",
      "uri": "string",
      "username": "string"
    }
  },
  "manifestCompression": enum,
  "ivSource": enum,
  "tsFileMode": enum,
  "manifestDurationFormat": enum,
  "keyFormatVersions": "string",
  "streamInfResolution": enum,
  "timestampDeltaMilliseconds": integer,
  "baseUrlContent": "string",
  "segmentationMode": enum,
  "captionLanguageMappings": [ {
    "languageDescription": "string",
    "captionChannel": integer,
    "languageCode": "string"
  } ],
  "clientCache": enum,
  "codecSpecification": enum,
  "keepSegments": integer,
  "redundantManifest": enum,
  "timedMetadataId3Period": integer,
  "programDateTime": enum,
  "directoryStructure": enum,
  "keyFormat": "string",
  "inputLossAction": enum,
  "adMarkers": [ enum
  ],
  "programDateTimePeriod": integer,
  "segmentLength": integer,
  "hlsCdnSettings": {
    "hlsAkamaiSettings": {
      "salt": "string",
      "httpTransferMode": enum,
      "numRetries": integer,
      "restartDelay": integer,
      "connectionRetryInterval": integer,
      "filecacheDuration": integer,
      "token": "string"
    },
    "hlsWebdavSettings": {
      "httpTransferMode": enum,
      "numRetries": integer,
      "restartDelay": integer,
      "connectionRetryInterval": integer,
      "filecacheDuration": integer
    },
    "hlsBasicPutSettings": {
      "numRetries": integer,
      "restartDelay": integer,
      "connectionRetryInterval": integer,
      "filecacheDuration": integer
    }
  }
}
"hlsMediaStoreSettings": {
"mediaStoreStorageClass": enum,
"numRetries": integer,
"restartDelay": integer,
"connectionRetryInterval": integer,
"fileCacheDuration": integer
},
"iFrameOnlyPlaylists": enum,
"frameCaptureGroupSettings": {
"destination": {
"destinationRefId": "string"
}
},
"name": "string"
],
"audioDescriptions": [{
"audioTypeControl": enum,
"languageCodeControl": enum,
"remixSettings": {
"channelMappings": [{
"outputChannel": integer,
"inputChannelLevels": [{
"inputChannel": integer,
"gain": integer
}
}
],
"channelsOut": integer,
"channelsIn": integer
},
"audioType": enum,
"name": "string",
"codecSettings": {
"aacSettings": {
"vbrQuality": enum,
"codingMode": enum,
"profile": enum,
"inputType": enum,
"bitrate": number,
"rawFormat": enum,
"rateControlMode": enum,
"sampleRate": number,
"spec": enum
},
"ac3Settings": {
"drcProfile": enum,
"dialnorm": integer,
"codingMode": enum,
"metadataControl": enum,
"bitrate": number,
"lfeFilter": enum,
"bitstreamMode": enum
},
"eac3Settings": {
"dialnorm": integer,
"passThroughControl": enum,
"drcLine": enum,
...
"metadataControl": enum,
"bitrate": number,
"ltRtSurroundMixLevel": number,
"surroundExMode": enum,
"lfeControl": enum,
"codingMode": enum,
"surroundMode": enum,
"attenuationControl": enum,
"lfeFilter": enum,
"dcFilter": enum,
"ltRtCenterMixLevel": number,
"phaseControl": enum,
"bitstreamMode": enum,
"stereoDownmix": enum,
"loRoSurroundMixLevel": number,
"drcRf": enum,
"loRoCenterMixLevel": number
},
"passThroughSettings": {
},
"mp2Settings": {
  "codingMode": enum,
  "bitrate": number,
  "sampleRate": number
},
"languageCode": "string",
"streamName": "string",
"audioNormalizationSettings": {
  "targetLkfs": number,
  "algorithmControl": enum,
  "algorithm": enum
},
"audioSelectorName": "string"
],
"availConfiguration": {
  "availSettings": {
    "scte35TimeSignalApos": {
      "adAvailOffset": integer,
      "webDeliveryAllowedFlag": enum,
      "noRegionalBlackoutFlag": enum
    },
    "scte35SpliceInsert": {
      "adAvailOffset": integer,
      "webDeliveryAllowedFlag": enum,
      "noRegionalBlackoutFlag": enum
    }
  }
},
"captionDescriptions": [
{
  "captionSelectorName": "string",
  "languageDescription": "string",
  "name": "string",
  "languageCode": "string",
  "destinationSettings": {
    "burnInDestinationSettings": {
      "xPosition": integer,
      "backgroundColor": enum,
      "yPosition": integer,
      "teletextGridControl": enum,
      "backgroundOpacity": integer,
      "fontOpacity": integer,
      "fontResolution": integer,
      "shadowOpacity": integer,
      "destinationControl": enum,
      "bitrate": number,
      "ltRtSurroundMixLevel": number,
      "surroundExMode": enum,
      "lfeControl": enum,
      "codingMode": enum,
      "surroundMode": enum,
      "attenuationControl": enum,
      "lfeFilter": enum,
      "dcFilter": enum,
      "ltRtCenterMixLevel": number,
      "phaseControl": enum,
      "bitstreamMode": enum,
      "stereoDownmix": enum,
      "loRoSurroundMixLevel": number,
      "drcRf": enum,
      "loRoCenterMixLevel": number
    }
  }
}]}
"shadowYOffset": integer,
"outlineSize": integer,
"outlineColor": enum,
"fontSize": "string",
"alignment": enum,
"shadowXOffset": integer,
"shadowColor": enum,
"fontColor": enum,
"font": {
  "passwordParam": "string",
  "uri": "string",
  "username": "string"
}

"scte27DestinationSettings": {

},

"teletextDestinationSettings": {

},

"ttmlDestinationSettings": {
  "styleControl": enum
},

"smpteTtDestinationSettings": {

},

"webvttDestinationSettings": {

},

"embeddedPlusScte20DestinationSettings": {

},

"dvbSubDestinationSettings": {
  "xPosition": integer,
  "backgroundColor": enum,
  "yPosition": integer,
  "teletextGridControl": enum,
  "backgroundColor": enum,
  "fontOpacity": integer,
  "fontResolution": integer,
  "shadowOpacity": integer,
  "shadowYOffset": integer,
  "outlineSize": integer,
  "outlineColor": enum,
  "fontSize": "string",
  "alignment": enum,
  "shadowXOffset": integer,
  "shadowColor": enum,
  "fontColor": enum,
  "font": {
    "passwordParam": "string",
    "uri": "string",
    "username": "string"
  }
},

"embeddedDestinationSettings": {

},

"rtmpCaptionInfoDestinationSettings": {

},

"aribDestinationSettings": {

},

"scte20PlusEmbeddedDestinationSettings": {

}

"globalConfiguration": {
  "inputLossBehavior": {
    "inputLossImageType": enum,
    "inputLossImageColor": "string",
    "inputLossImageSlate": {

    }
  }
}
}
"passwordParam": "string",
"uri": "string",
"username": "string"
},
"repeatFrameMsec": integer,
"blackFrameMsec": integer
},
"supportLowFramerateInputs": enum,
"outputLockingMode": enum,
"initialAudioGain": integer,
"inputEndAction": enum,
"outputTimingSource": enum
},
"videoDescriptions": [
{
"respondToAfd": enum,
"scalingBehavior": enum,
"name": "string",
"width": integer,
"sharpness": integer,
"codecSettings": {
"h264Settings": {
"minIInterval": integer,
"slices": integer,
"parNumerator": integer,
"gopsSizeUnits": enum,
"subgopLength": enum,
"maxBitrate": integer,
"bitrate": integer,
"bufFillPct": integer,
"temporalAq": enum,
"afdSignaling": enum,
"timecodeInsertion": enum,
"bufSize": integer,
"softness": integer,
"framerateControl": enum,
"gvrbrQualityLevel": integer,
"fixedAfd": enum,
"level": enum,
"lookAheadRateControl": enum,
"profile": enum,
"framerateNumerator": integer,
"gopsClosedCadence": integer,
"entropyEncoding": enum,
"framerateDenominator": integer,
"spatialAq": enum,
"adaptiveQuantization": enum,
"colorMetadata": enum,
"gopsSize": number,
"numRefFrames": integer,
"gopBReference": enum,
"parControl": enum,
"parDenominator": integer,
"syntax": enum,
"sceneChangeDetect": enum,
"scanType": enum,
"flickerAq": enum,
"gopNumBFrames": integer,
"rateControlMode": enum
},
"frameCaptureSettings": {
"captureInterval": integer
}
},
"height": integer
}
Schemas

```
"availBlanking": {
  "state": enum,
  "availBlankingImage": {
    "passwordParam": "string",
    "uri": "string",
    "username": "string"
  }
},
"blackoutSlate": {
  "networkEndBlackoutImage": {
    "passwordParam": "string",
    "uri": "string",
    "username": "string"
  },
  "networkEndBlackout": enum,
  "networkId": "string",
  "state": enum,
  "blackoutSlateImage": {
    "passwordParam": "string",
    "uri": "string",
    "username": "string"
  }
},
"egressEndpoints": [
  {
    "sourceIp": "string"
  }
],
"inputSpecification": {
  "codec": enum,
  "resolution": enum,
  "maximumBitrate": enum
},
"channelClass": enum,
"tags": {
},
"logLevel": enum,
"roleArn": "string",
"name": "string",
"id": "string",
"state": enum,
"pipelinesRunningCount": integer,
"arn": "string"
```

Example InvalidRequest

```
{
  "message": "string"
}
```

Example AccessDenied

```
{
  "message": "string"
}
```

Example ResourceNotFound

```
{
}
```
Properties

**AacCodingMode**

Aac Coding Mode

- AD_RECEIVER_MIX
- CODING_MODE_1_0
- CODING_MODE_1_1
- CODING_MODE_2_0
- CODING_MODE_5_1

**AacInputType**

Aac Input Type

- BROADCASTER_MIXED_AD
- NORMAL
AacProfile

Aac Profile

- HEV1
- HEV2
- LC

AacRateControlMode

Aac Rate Control Mode

- CBR
- VBR

AacRawFormat

Aac Raw Format

- LATM LOAS
- NONE

AacSettings

Aac Settings

vbrQuality

VBR Quality Level - Only used if rateControlMode is VBR.

  Type: AacVbrQuality (p. 446)
  Required: False

codingMode

Mono, Stereo, or 5.1 channel layout. Valid values depend on rate control mode and profile. The adReceiverMix 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.

  Type: AacCodingMode (p. 444)
  Required: False

profile

AAC Profile.

  Type: AacProfile (p. 445)
  Required: False

inputType

Set to "broadcasterMixedAd" when input contains pre-mixed main audio + AD (narration) as a stereo pair. The Audio Type field (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. The values in audioTypeControl and audioType (in AudioDescription) are ignored when set to broadcasterMixedAd. Leave set to "normal" when input does not contain pre-mixed audio + AD.

- **Type**: AacInputType (p. 444)
- **Required**: False

### bitrate

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

- **Type**: number
- **Required**: False

### rawFormat

Sets LATM / LOAS AAC output for raw containers.

- **Type**: AacRawFormat (p. 445)
- **Required**: False

### rateControlMode

Rate Control Mode.

- **Type**: AacRateControlMode (p. 445)
- **Required**: False

### sampleRate

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

- **Type**: number
- **Required**: False

### spec

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

- **Type**: AacSpec (p. 446)
- **Required**: False

### AacSpec

Aac Spec
- MPEG2
- MPEG4

### AacVbrQuality

Aac Vbr Quality
HIGH
LOW
MEDIUM_HIGH
MEDIUM_LOW

**Ac3BitstreamMode**

Ac3 Bitstream Mode

COMMENTARY
COMPLETE_MAIN
DIALOGUE
EMERGENCY
HEARING_IMPAIRED
MUSIC_AND_EFFECTS
VISUALLY_IMPAIRED
VOICE_OVER

**Ac3CodingMode**

Ac3 Coding Mode

CODING_MODE_1_0
CODING_MODE_1_1
CODING_MODE_2_0
CODING_MODE_3_2_LFE

**Ac3DrcProfile**

Ac3 Drc Profile

FILM_STANDARD
NONE

**Ac3LfeFilter**

Ac3 Lfe Filter

DISABLED
ENABLED

**Ac3MetadataControl**

Ac3 Metadata Control

FOLLOW_INPUT
USE_CONFIGURED

**Ac3Settings**

Ac3 Settings
**drcProfile**

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

  Type: Ac3DrcProfile (p. 447)  
  Required: False

**dialnorm**

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

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

**codingMode**

Dolby Digital coding mode. Determines number of channels.

  Type: Ac3CodingMode (p. 447)  
  Required: False

**metadataControl**

When set to "followInput", 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.

  Type: Ac3MetadataControl (p. 447)  
  Required: False

**bitrate**

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

  Type: number  
  Required: False

**lfeFilter**

When set to enabled, applies a 120Hz lowpass filter to the LFE channel prior to encoding. Only valid in codingMode32Lfe mode.

  Type: Ac3LfeFilter (p. 447)  
  Required: False

**bitstreamMode**

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

  Type: Ac3BitstreamMode (p. 447)
Required: False

AccessDenied

message

Type: string
Required: False

AfdSignaling

Afd Signaling

AUTO
FIXED
NONE

ArchiveContainerSettings

Archive Container Settings

m2tsSettings

Type: M2tsSettings (p. 518)
Required: False

ArchiveGroupSettings

Archive Group Settings

destination

A directory and base filename where archive files should be written.

Type: OutputLocationRef (p. 536)
Required: True

rolloverInterval

Number of seconds to write to archive file before closing and starting a new one.

Type: integer
Required: False
Minimum: 1

ArchiveOutputSettings

Archive Output Settings

extension

Output file extension. If excluded, this will be auto-selected from the container type.
**containerSettings**
Settings specific to the container type of the file.

- **Type**: `ArchiveContainerSettings` (p. 449)
- **Required**: True

**nameModifier**
String concatenated to the end of the destination filename. Required for multiple outputs of the same type.

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

**AribDestinationSettings**
Arib Destination Settings

**AribSourceSettings**
Arib Source Settings

**AudioChannelMapping**
Audio Channel Mapping

**outputChannel**
The index of the output channel being produced.

- **Type**: integer
- **Required**: True
- **Minimum**: 0
- **Maximum**: 7

**inputChannelLevels**
Indices and gain values for each input channel that should be remixed into this output channel.

- **Type**: Array of type `InputChannelLevel` (p. 509)
- **Required**: True

**AudioCodecSettings**
Audio Codec Settings

**aacSettings**

- **Type**: `AacSettings` (p. 445)
- **Required**: False
**ac3Settings**

- **Type:** Ac3Settings (p. 447)
- **Required:** False

**eac3Settings**

- **Type:** Eac3Settings (p. 476)
- **Required:** False

**passThroughSettings**

- **Type:** PassThroughSettings (p. 537)
- **Required:** False

**mp2Settings**

- **Type:** Mp2Settings (p. 529)
- **Required:** False

**AudioDescription**

Audio Description

**audioTypeControl**

Determines how audio type is determined. followInput: If the input contains an ISO 639 audioType, then that value is passed through to the output. If the input contains no ISO 639 audioType, the value in Audio Type is included in the output. useConfigured: The value in Audio Type is included in the output. Note that this field and audioType are both ignored if inputType is broadcasterMixedAd.

- **Type:** AudioDescriptionAudioTypeControl (p. 452)
- **Required:** False

**languageCodeControl**

Choosing followInput will cause the ISO 639 language code of the output to follow the ISO 639 language code of the input. The languageCode will be used when useConfigured is set, or when followInput is selected but there is no ISO 639 language code specified by the input.

- **Type:** AudioDescriptionLanguageCodeControl (p. 453)
- **Required:** False

**remixSettings**

Settings that control how input audio channels are remixed into the output audio channels.

- **Type:** RemixSettings (p. 537)
- **Required:** False

**audioType**

Applies only if audioTypeControl is useConfigured. The values for audioType are defined in ISO-IEC 13818-1.
Properties

name

The name of this AudioDescription. Outputs will use this name to uniquely identify this AudioDescription. Description names should be unique within this Live Event.

Type: string
Required: True

codecSettings

Audio codec settings.

Type: AudioCodecSettings (p. 450)
Required: False

languageCode

Indicates the language of the audio output track. Only used if languageControlMode is useConfigured, or there is no ISO 639 language code specified in the input.

Type: string
Required: False
MinLength: 3
MaxLength: 3

streamName

Used for MS Smooth and Apple HLS outputs. Indicates the name displayed by the player (eg. English, or Director Commentary).

Type: string
Required: False

audioNormalizationSettings

Advanced audio normalization settings.

Type: AudioNormalizationSettings (p. 454)
Required: False

audioSelectorName

The name of the AudioSelector used as the source for this AudioDescription.

Type: string
Required: True

AudioDescriptionAudioTypeControl

Audio Description Audio Type Control
FOLLOW_INPUT
USE_CONFIGURED

**AudioDescriptionLanguageCodeControl**

Audio Description Language Code Control

FOLLOW_INPUT
USE_CONFIGURED

**AudioLanguageSelection**

Audio Language Selection

**languageSelectionPolicy**

When set to "strict", the transport stream demux strictly identifies audio streams by their language descriptor. If a PMT update occurs such that an audio stream matching the initially selected language is no longer present then mute will be encoded until the language returns. If "loose", then on a PMT update the demux will choose another audio stream in the program with the same stream type if it can't find one with the same language.

*Type: AudioLanguageSelectionPolicy (p. 453)*

*Required: False*

**languageCode**

Selects a specific three-letter language code from within an audio source.

*Type: string*

*Required: True*

**AudioLanguageSelectionPolicy**

Audio Language Selection Policy

LOOSE
STRICT

**AudioNormalizationAlgorithm**

Audio Normalization Algorithm

ITU_1770_1
ITU_1770_2

**AudioNormalizationAlgorithmControl**

Audio Normalization Algorithm Control

CORRECT_AUDIO
AudioNormalizationSettings

Audio Normalization Settings

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  
Minimum: -59.0  
Maximum: 0.0

algorithmControl

When set to correctAudio the output audio is corrected using the chosen algorithm. If set to measureOnly, the audio will be measured but not adjusted.

Type: AudioNormalizationAlgorithmControl (p. 453)  
Required: False

algorithm

Audio normalization algorithm to use. itu17701 conforms to the CALM Act specification, itu17702 conforms to the EBU R-128 specification.

Type: AudioNormalizationAlgorithm (p. 453)  
Required: False

AudioOnlyHlsSettings

Audio Only Hls Settings

audioTrackType

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

Type: AudioOnlyHlsTrackType (p. 455)  
Required: False

audioGroupId

Specifies the group to which the audio Rendition belongs.

Type: string
Properties

**Required**: False

**audioOnlyImage**

For use with an audio only Stream. Must be a .jpg or .png file. If given, this image will be used as the cover-art for the audio only output. Ideally, it should be formatted for an iPhone screen for two reasons. The iPhone does not resize the image, it crops a centered image on the top/bottom and left/right. Additionally, this image file gets saved bit-for-bit into every 10-second segment file, so will increase bandwidth by \{image file size\} * \{segment count\} * \{user count\}.

**Type**: InputLocation (p. 510)

**Required**: False

**AudioOnlyHlsTrackType**

Audio Only Hls Track Type

ALTERNATE_AUDIO_AUTO_SELECT
ALTERNATE_AUDIO_AUTO_SELECT_DEFAULT
ALTERNATE_AUDIO_NOT_AUTO_SELECT
AUDIO_ONLY_VARIANT_STREAM

**AudioPidSelection**

Audio Pid Selection

**pid**

Selects a specific PID from within a source.

**Type**: integer

**Required**: True

**Minimum**: 0

**Maximum**: 8191

**AudioSelector**

Audio Selector

**name**

The name of this AudioSelector. AudioDescriptions will use this name to uniquely identify this Selector. Selector names should be unique per input.

**Type**: string

**Required**: True

**MinLength**: 1

**selectorSettings**

The audio selector settings.

**Type**: AudioSelectorSettings (p. 456)

**Required**: False
**AudioSelectorSettings**

Audio Selector Settings

**audioLanguageSelection**

- **Type:** AudioLanguageSelection (p. 453)
- **Required:** False

**audioPidSelection**

- **Type:** AudioPidSelection (p. 455)
- **Required:** False

**AudioType**

Audio Type

- CLEAN_EFFECTS
- HEARING_IMPAIRED
- UNDEFINED
- VISUAL_IMPAIRED_COMMENTARY

**AuthenticationScheme**

Authentication Scheme

- AKAMAI
- COMMON

**AvailBlanking**

Avail Blanking

**state**

When set to enabled, causes video, audio and captions to be blanked when insertion metadata is added.

- **Type:** AvailBlankingState (p. 456)
- **Required:** False

**availBlankingImage**

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

- **Type:** InputLocation (p. 510)
- **Required:** False

**AvailBlankingState**

Avail Blanking State
DISABLED
ENABLED

**AvailConfiguration**

Avail Configuration

**availSettings**

Ad avail settings.

*Type:* AvailSettings (p. 457)

*Required:* False

**AvailSettings**

Avail Settings

**scte35TimeSignalApos**

*Type:* Scte35TimeSignalApos (p. 542)

*Required:* False

**scte35SpliceInsert**

*Type:* Scte35SpliceInsert (p. 541)

*Required:* False

**BadGatewayException**

**message**

*Type:* string

*Required:* False

**BlackoutSlate**

Blackout Slate

**networkEndBlackoutImage**

Path to local file to use as Network End Blackout image. Image will be scaled to fill the entire output raster.

*Type:* InputLocation (p. 510)

*Required:* False

**networkEndBlackout**

Setting to enabled causes the encoder to blackout the video, audio, and captions, and raise the "Network Blackout Image" slate when an SCTE104/35 Network End Segmentation Descriptor is encountered.
The blackout will be lifted when the Network Start Segmentation Descriptor is encountered. The Network End and Network Start descriptors must contain a network ID that matches the value entered in "Network ID".

**Type:** BlackoutSlateNetworkEndBlackout (p. 458)
**Required:** False

**networkId**
Provides Network ID that matches EIDR ID format (e.g., "10.XXXX/XXXX-XXXX-XXXX-XXXX-XXXX-C").

**Type:** string
**Required:** False
**MinLength:** 34
**MaxLength:** 34

**state**
When set to enabled, causes video, audio and captions to be blanked when indicated by program metadata.

**Type:** BlackoutSlateState (p. 458)
**Required:** False

**blackoutSlateImage**
Blackout slate image to be used. Leave empty for solid black. Only bmp and png images are supported.

**Type:** InputLocation (p. 510)
**Required:** False

**BlackoutSlateNetworkEndBlackout**
Blackout Slate Network End Blackout

- DISABLED
- ENABLED

**BlackoutSlateState**
Blackout Slate State

- DISABLED
- ENABLED

**BurnInAlignment**
Burn In Alignment

- CENTERED
- LEFT
- SMART
**BurnInBackgroundColor**

Burn In Background Color

- BLACK
- NONE
- WHITE

**BurnInDestinationSettings**

Burn In Destination Settings

**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 xPosition is provided, the horizontal caption position will be determined by the alignment parameter. All burn-in and DVB-Sub font settings must match.

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

**backgroundColor**

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

- **Type**: BurnInBackgroundColor (p. 459)
- **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 yPosition is provided, the caption will be positioned towards the bottom of the output. All burn-in and DVB-Sub font settings must match.

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

**teletextGridControl**

Controls whether a fixed grid size will be used to generate the output subtitles bitmap. Only applicable for Teletext inputs and DVB-Sub/Burn-in outputs.

- **Type**: BurnInTeletextGridControl (p. 462)
- **Required**: False

**backgroundOpacity**

Specifies the opacity of the background rectangle. 255 is opaque; 0 is transparent. Leaving this parameter out 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: False
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 out 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

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: 0
Properties

Maximum: 10

outlineColor
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.

  Type: BurnInOutlineColor (p. 462)
  Required: False

fontSize
When set to 'auto' fontSize will scale depending on the size of the output. Giving a positive integer will specify the exact font size in points. All burn-in and DVB-Sub font settings must match.

  Type: string
  Required: False

alignment
If no explicit xPosition or yPosition 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. Selecting "smart" justification will left-justify live subtitles and center-justify pre-recorded subtitles. All burn-in and DVB-Sub font settings must match.

  Type: BurnInAlignment (p. 458)
  Required: False

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

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

  Type: BurnInShadowColor (p. 462)
  Required: False

fontColor
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.

  Type: BurnInFontColor (p. 462)
  Required: False
### font

External font file used for caption burn-in. File extension must be 'ttf' or 'tte'. Although the user can select output fonts for many different types of input captions, embedded, STL and teletext sources use a strict grid system. Using external fonts with these caption sources could cause unexpected display of proportional fonts. All burn-in and DVB-Sub font settings must match.

- **Type**: `InputLocation` (p. 510)
- **Required**: False

#### BurnInFontColor

Burn In Font Color

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

#### BurnInOutlineColor

Burn In Outline Color

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

#### BurnInShadowColor

Burn In Shadow Color

- BLACK
- NONE
- WHITE

#### BurnInTeletextGridControl

Burn In Teletext Grid Control

- FIXED
- SCALED

#### CaptionDescription

Output groups for this Live Event. Output groups contain information about where streams should be distributed.
captionSelectorName

Specifies which input caption selector to use as a caption source when generating output captions. This field should match a captionSelector name.

Type: string
Required: True

languageDescription

Human readable information to indicate captions available for players (eg. English, or Spanish).

Type: string
Required: False

name

Name of the caption description. Used to associate a caption description with an output. Names must be unique within an event.

Type: string
Required: True

languageCode


Type: string
Required: False

destinationSettings

Additional settings for captions destination that depend on the destination type.

Type: CaptionDestinationSettings (p. 463)
Required: False

CaptionDestinationSettings

Caption Destination Settings

burnInDestinationSettings

Type: BurnInDestinationSettings (p. 459)
Required: False

scte27DestinationSettings

Type: Scte27DestinationSettings (p. 541)
Required: False

teletextDestinationSettings

Type: TeletextDestinationSettings (p. 545)
Required: False

**ttmlDestinationSettings**

*Type:* [TtmlDestinationSettings](#)  
*Required:* False

**smpteTtDestinationSettings**

*Type:* [SmpteTtDestinationSettings](#)  
*Required:* False

**webvttDestinationSettings**

*Type:* [WebvttDestinationSettings](#)  
*Required:* False

**embeddedPlusScte20DestinationSettings**

*Type:* [EmbeddedPlusScte20DestinationSettings](#)  
*Required:* False

**dvbSubDestinationSettings**

*Type:* [DvbSubDestinationSettings](#)  
*Required:* False

**embeddedDestinationSettings**

*Type:* [EmbeddedDestinationSettings](#)  
*Required:* False

**rtmpCaptionInfoDestinationSettings**

*Type:* [RtmpCaptionInfoDestinationSettings](#)  
*Required:* False

**aribDestinationSettings**

*Type:* [AribDestinationSettings](#)  
*Required:* False

**scte20PlusEmbeddedDestinationSettings**

*Type:* [Scte20PlusEmbeddedDestinationSettings](#)  
*Required:* False

**CaptionLanguageMapping**

Maps a caption channel to an ISO 693-2 language code (http://www.loc.gov/standards/iso639-2), with an optional description.
languageDescription

Textual description of language

Type: string
Required: True
MinLength: 1

captionChannel

The closed caption channel being described by this CaptionLanguageMapping. Each channel mapping must have a unique channel number (maximum of 4)

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

languageCode

Three character ISO 639-2 language code (see http://www.loc.gov/standards/iso639-2)

Type: string
Required: True
MinLength: 3
MaxLength: 3

CaptionSelector

Output groups for this Live Event. Output groups contain information about where streams should be distributed.

name

Name identifier for a caption selector. This name is used to associate this caption selector with one or more caption descriptions. Names must be unique within an event.

Type: string
Required: True
MinLength: 1

languageCode

When specified this field indicates the three letter language code of the caption track to extract from the source.

Type: string
Required: False

selectorSettings

Caption selector settings.

Type: CaptionSelectorSettings (p. 466)
Required: False

CaptionSelectorSettings
Caption Selector Settings

embeddedSourceSettings
Type: EmbeddedSourceSettings (p. 480)
Required: False

scte20SourceSettings
Type: Scte20SourceSettings (p. 540)
Required: False

dvbSubSourceSettings
Type: DvbSubSourceSettings (p. 474)
Required: False

teletextSourceSettings
Type: TeletextSourceSettings (p. 545)
Required: False

aribSourceSettings
Type: AribSourceSettings (p. 450)
Required: False

scte27SourceSettings
Type: Scte27SourceSettings (p. 541)
Required: False

Channel

inputAttachments
List of input attachments for channel.
Type: Array of type InputAttachment (p. 509)
Required: False

destinations
A list of destinations of the channel. For UDP outputs, there is one destination per output. For other types (HLS, for example), there is one destination per packager.
Type: Array of type OutputDestination (p. 534)
encoderSettings
Type: EncoderSettings (p. 481)
Required: False

egressEndpoints
The endpoints where outgoing connections initiate from
Type: Array of type ChannelEgressEndpoint (p. 468)
Required: False

inputSpecification
Type: InputSpecification (p. 514)
Required: False

channelClass
The class for this channel. STANDARD for a channel with two pipelines or SINGLE_PIPELINE for a channel with one pipeline.
Type: ChannelClass (p. 468)
Required: False

tags
A collection of key-value pairs.
Type: Tags (p. 544)
Required: False

logLevel
The log level being written to CloudWatch Logs.
Type: LogLevel (p. 515)
Required: False

roleArn
The Amazon Resource Name (ARN) of the role assumed when running the Channel.
Type: string
Required: False

name
The name of the channel. (user-mutable)
Type: string
Required: False
id
The unique id of the channel.
  Type: string
  Required: False

state
  Type: ChannelState (p. 468)
  Required: False

pipelinesRunningCount
The number of currently healthy pipelines.
  Type: integer
  Required: False

arn
The unique arn of the channel.
  Type: string
  Required: False

ChannelClass
A standard channel has two encoding pipelines and a single pipeline channel only has one.
  STANDARD
  SINGLE_PIPELINE

ChannelEgressEndpoint
sourceIp
Public IP of where a channel's output comes from
  Type: string
  Required: False

ChannelState
  CREATING
  CREATE_FAILED
  IDLE
  STARTING
  RUNNING
  RECOVERING
  STOPPING
  DELETING
  DELETED
DvbNitSettings

DVB Network Information Table (NIT)

**networkName**

The network name text placed in the networkNameDescriptor inside the Network Information Table. Maximum length is 256 characters.

*Type: string*
*Required: True*
*MinLength: 1*
*MaxLength: 256*

**networkId**

The numeric value placed in the Network Information Table (NIT).

*Type: integer*
*Required: True*
*Minimum: 0*
*Maximum: 65536*

**repInterval**

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

*Type: integer*
*Required: False*
*Minimum: 25*
*Maximum: 10000*

DvbSdtOutputSdt

Dvb Sdt Output Sdt

**DvbSdtSettings**

DVB Service Description Table (SDT)

**serviceName**

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

*Type: string*
**Properties**

**Required**: False
**MinLength**: 1
**MaxLength**: 256

**serviceProviderName**

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

**Type**: string
**Required**: False
**MinLength**: 1
**MaxLength**: 256

**repInterval**

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

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

**outputSdt**

Selects method of inserting SDT information into output stream. The sdtFollow setting copies SDT information from input stream to output stream. The sdtFollowIfPresent setting 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. The sdtManual setting means user will enter the SDT information. The sdtNone setting means output stream will not contain SDT information.

**Type**: DvbSdtOutputSdt (p. 469)
**Required**: False

**DvbSubDestinationAlignment**

Dvb Sub Destination Alignment

CENTERED
LEFT
SMART

**DvbSubDestinationBackgroundColor**

Dvb Sub Destination Background Color

BLACK
NONE
WHITE

**DvbSubDestinationFontColor**

Dvb Sub Destination Font Color
**DvbSubDestinationOutlineColor**

Dvb Sub Destination Outline Color

BLACK
BLUE
GREEN
RED
WHITE
YELLOW

**DvbSubDestinationSettings**

Dvb Sub Destination Settings

**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 xPosition 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

**backgroundColor**

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

- **Type:** DvbSubDestinationBackgroundColor (p. 470)
- **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 yPosition 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
teletextGridControl

Controls whether a fixed grid size will be used to generate the output subtitles bitmap. Only applicable for Teletext inputs and DVB-Sub/Burn-in outputs.

Type: DvbSubDestinationTeletextGridControl (p. 474)
Required: False

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: False
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
outlineSize
Specifies font outline size in pixels. This option is not valid for source captions that are either 608/embrbed 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**: 10

outlineColor
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.

- **Type**: DvbSubDestinationOutlineColor (p. 471)
- **Required**: False

fontSize
When set to auto fontSize will scale depending on the size of the output. Giving a positive integer will specify the exact font size in points. All burn-in and DVB-Sub font settings must match.

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

alignment
If no explicit xPosition or yPosition 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. Selecting "smart" justification will left-justify live subtitles and center-justify pre-recorded subtitles. This option is not valid for source captions that are STL or 608/embedded. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- **Type**: DvbSubDestinationAlignment (p. 470)
- **Required**: False

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

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

- **Type**: DvbSubDestinationShadowColor (p. 474)
- **Required**: False
**fontColor**

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.

**Type:** DvbSubDestinationFontColor (p. 470)
**Required:** False

**font**

External font file used for caption burn-in. File extension must be 'ttf' or 'tte'. Although the user can select output fonts for many different types of input captions, embedded, STL and teletext sources use a strict grid system. Using external fonts with these caption sources could cause unexpected display of proportional fonts. All burn-in and DVB-Sub font settings must match.

**Type:** InputLocation (p. 510)
**Required:** False

**DvbSubDestinationShadowColor**

Dvb Sub Destination Shadow Color

BLACK
NONE
WHITE

**DvbSubDestinationTeletextGridControl**

Dvb Sub Destination Teletext Grid Control

FIXED
SCALED

**DvbSubSourceSettings**

Dvb Sub Source Settings

**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

**DvbTdtSettings**

DVB Time and Date Table (SDT)

**repInterval**

The number of milliseconds between instances of this table in the output transport stream.
**Type**: integer  
**Required**: False  
**Minimum**: 1000  
**Maximum**: 30000

**Eac3AttenuationControl**

Eac3 Attenuation Control  
ATTENUATE_3_DB  
NONE

**Eac3BitstreamMode**

Eac3 Bitstream Mode  
COMMENTARY  
COMPLETE_MAIN  
EMERGENCY  
HEARING_IMPAIRED  
VISUALLY_IMPAIRED

**Eac3CodingMode**

Eac3 Coding Mode  
CODING_MODE_1_0  
CODING_MODE_2_0  
CODING_MODE_3_2

**Eac3DcFilter**

Eac3 Dc Filter  
DISABLED  
ENABLED

**Eac3DrcLine**

Eac3 Drc Line  
FILM_LIGHT  
FILM_STANDARD  
MUSIC_LIGHT  
MUSIC_STANDARD  
NONE  
SPEECH

**Eac3DrcRf**

Eac3 Drc Rf
FILM_LIGHT
FILM_STANDARD
MUSIC_LIGHT
MUSIC_STANDARD
NONE
SPEECH

**Eac3LfeControl**

Eac3 Lfe Control

- LFE
- NO_LFE

**Eac3LfeFilter**

Eac3 Lfe Filter

- DISABLED
- ENABLED

**Eac3MetadataControl**

Eac3 Metadata Control

- FOLLOW_INPUT
- USE_CONFIGURED

**Eac3PassthroughControl**

Eac3 Passthrough Control

- NO_PASSTHROUGH
- WHEN_POSSIBLE

**Eac3PhaseControl**

Eac3 Phase Control

- NO_SHIFT
- SHIFT_90_DEGREES

**Eac3Settings**

Eac3 Settings

**dialnorm**

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

*Type:* integer
Properties

Required: False
Minimum: 1
Maximum: 31

**passthroughControl**

When set to whenPossible, 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.

*Type: Eac3PassthroughControl (p. 476)*
*Required: False*

**drcLine**

Sets the Dolby dynamic range compression profile.

*Type: Eac3DrcLine (p. 475)*
*Required: False*

**metadataControl**

When set to followInput, 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.

*Type: Eac3MetadataControl (p. 476)*
*Required: False*

**bitrate**

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

*Type: number*
*Required: False*

**ltRtSurroundMixLevel**

Left total/Right total surround mix level. Only used for 3/2 coding mode.

*Type: number*
*Required: False*

**surroundExMode**

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

*Type: Eac3SurroundExMode (p. 479)*
*Required: False*

**lfeControl**

When encoding 3/2 audio, setting to lfe enables the LFE channel
Type: Eac3LfeControl (p. 476)
Required: False

codingMode
Dolby Digital Plus coding mode. Determines number of channels.
Type: Eac3CodingMode (p. 475)
Required: False

surroundMode
When encoding 2/0 audio, sets whether Dolby Surround is matrix encoded into the two channels.
Type: Eac3SurroundMode (p. 480)
Required: False

attenuationControl
When set to attenuate3Db, applies a 3 dB attenuation to the surround channels. Only used for 3/2 coding mode.
Type: Eac3AttenuationControl (p. 475)
Required: False

lfeFilter
When set to enabled, applies a 120Hz lowpass filter to the LFE channel prior to encoding. Only valid with codingMode32 coding mode.
Type: Eac3LfeFilter (p. 476)
Required: False

dcFilter
When set to enabled, activates a DC highpass filter for all input channels.
Type: Eac3DcFilter (p. 475)
Required: False

ltRtCenterMixLevel
Left total/Right total center mix level. Only used for 3/2 coding mode.
Type: number
Required: False

phaseControl
When set to shift90Degrees, applies a 90-degree phase shift to the surround channels. Only used for 3/2 coding mode.
Type: Eac3PhaseControl (p. 476)
Required: False

bitstreamMode

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

Type: Eac3BitstreamMode (p. 475)
Required: False

stereoDownmix

Stereo downmix preference. Only used for 3/2 coding mode.

Type: Eac3StereoDownmix (p. 479)
Required: False

loRoSurroundMixLevel

Left only/Right only surround mix level. Only used for 3/2 coding mode.

Type: number
Required: False

drcRf

Sets the profile for heavy Dolby dynamic range compression, ensures that the instantaneous signal peaks do not exceed specified levels.

Type: Eac3DrcRf (p. 475)
Required: False

loRoCenterMixLevel

Left only/Right only center mix level. Only used for 3/2 coding mode.

Type: number
Required: False

Eac3StereoDownmix

Eac3 Stereo Downmix

DPL2
LO_RO
LT_RT
NOT_INDICATED

Eac3SurroundExMode

Eac3 Surround Ex Mode

DISABLED
ENABLED
NOT_INDICATED

**Eac3SurroundMode**

Eac3 Surround Mode

DISABLED
ENABLED
NOT_INDICATED

**EmbeddedConvert608To708**

Embedded Convert608 To708

DISABLED
UPCONVERT

**EmbeddedDestinationSettings**

Embedded Destination Settings

**EmbeddedPlusScte20DestinationSettings**

Embedded Plus Scte20 Destination Settings

**EmbeddedScte20Detection**

Embedded Scte20 Detection

AUTO
OFF

**EmbeddedSourceSettings**

Embedded Source Settings

**scte20Detection**

Set to "auto" to handle streams with intermittent and/or non-aligned SCTE-20 and Embedded captions.

*Type: EmbeddedScte20Detection (p. 480)*

*Required: False*

**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**

If 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.

Type: EmbeddedConvert608To708 (p. 480)
Required: False

**source608TrackNumber**

This field is unused and deprecated.

Type: integer
Required: False
Minimum: 1
Maximum: 5

**EncoderSettings**

Encoder Settings

**timecodeConfig**

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

Type: TimecodeConfig (p. 545)
Required: True

**outputGroups**

Type: Array of type OutputGroup (p. 535)
Required: True

**audioDescriptions**

Type: Array of type AudioDescription (p. 451)
Required: True

**availConfiguration**

Event-wide configuration settings for ad avail insertion.

Type: AvailConfiguration (p. 457)
Required: False

**captionDescriptions**

Settings for caption descriptions

Type: Array of type CaptionDescription (p. 462)
Required: False
**globalConfiguration**

Configuration settings that apply to the event as a whole.

- **Type:** GlobalConfiguration (p. 484)
- **Required:** False

**videoDescriptions**

- **Type:** Array of type VideoDescription (p. 548)
- **Required:** True

**availBlanking**

Settings for ad avail blanking.

- **Type:** AvailBlanking (p. 456)
- **Required:** False

**blackoutSlate**

Settings for blackout slate.

- **Type:** BlackoutSlate (p. 457)
- **Required:** False

**FecOutputIncludeFec**

Fec Output Include Fec

- COLUMN
- COLUMN_AND_ROW

**FecOutputSettings**

Fec Output Settings

**rowLength**

Parameter L from SMPTE 2022-1. The width of the FEC protection matrix. Must be between 1 and 20, inclusive. If only Column FEC is used, then larger values increase robustness. If Row FEC is used, then this is the number of transport stream packets per row error correction packet, and the value must be between 4 and 20, inclusive, if includeFec is columnAndRow. If includeFec is column, this value must be 1 to 20, inclusive.

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 20

**columnDepth**

Parameter D from SMPTE 2022-1. The height of the FEC protection matrix. The number of transport stream packets per column error correction packet. Must be between 4 and 20, inclusive.
**Type**: integer  
**Required**: False  
**Minimum**: 4  
**Maximum**: 20

**includeFec**

Enables column only or column and row based FEC

**Type**: FecOutputIncludeFec (p. 482)  
**Required**: False

**FixedAfd**

Fixed Afd

- AFD_0000
- AFD_0010
- AFD_0011
- AFD_0100
- AFD_1000
- AFD_1001
- AFD_1010
- AFD_1011
- AFD_1101
- AFD_1110
- AFD_1111

**FrameCaptureGroupSettings**

Frame Capture Group Settings

**destination**

The destination for the frame capture files. Either the URI for an Amazon S3 bucket and object, plus a file name prefix (for example, s3ssl://sportsDelivery/highlights/20180820/curling_) or the URI for a MediaStore container, plus a file name prefix (for example, mediastoressl://sportsDelivery/20180820/curling_). The final file names consist of the prefix from the destination field (for example, "curling_") + name modifier + the counter (5 digits, starting from 00001) + extension (which is always .jpg). For example, curlingLow.00001.jpg

**Type**: OutputLocationRef (p. 536)  
**Required**: True

**FrameCaptureOutputSettings**

Frame Capture Output Settings

**nameModifier**

Required if the output group contains more than one output. This modifier forms part of the output file name.

**Type**: string
Required: False

FrameCaptureSettings
Frame Capture Settings

captureInterval
The frequency, in seconds, for capturing frames for inclusion in the output. For example, "10" means capture a frame every 10 seconds.

Type: integer
    Required: True
    Minimum: 1
    Maximum: 3600

GatewayTimeoutException

message

Type: string
    Required: False

GlobalConfiguration
Global Configuration

inputLossBehavior
Settings for system actions when input is lost.

Type: InputLossBehavior (p. 511)
    Required: False

supportLowFramerateInputs
Adjusts video input buffer for streams with very low video framerates. This is commonly set to enabled for music channels with less than one video frame per second.

Type: GlobalConfigurationLowFramerateInputs (p. 485)
    Required: False

outputLockingMode
Indicates how MediaLive pipelines are synchronized. PIPELINELOCKING - MediaLive will attempt to synchronize the output of each pipeline to the other. EPOCHLOCKING - MediaLive will attempt to synchronize the output of each pipeline to the Unix epoch.

Type: GlobalConfigurationOutputLockingMode (p. 485)
    Required: False

initialAudioGain
Value to set the initial audio gain for the Live Event.
**Properties**

**Type**: integer  
**Required**: False  
**Minimum**: -60  
**Maximum**: 60

**inputEndAction**

Indicates the action to take when the current input completes (e.g. end-of-file). When switchAndLoopInputs is configured the encoder will restart at the beginning of the first input. When "none" is configured the encoder will transcode either black, a solid color, or a user specified slate images per the "Input Loss Behavior" configuration until the next input switch occurs (which is controlled through the Channel Schedule API).

**Type**: GlobalConfigurationInputEndAction (p. 485)  
**Required**: False

**outputTimingSource**

Indicates whether the rate of frames emitted by the Live encoder should be paced by its system clock (which optionally may be locked to another source via NTP) or should be locked to the clock of the source that is providing the input stream.

**Type**: GlobalConfigurationOutputTimingSource (p. 485)  
**Required**: False

**GlobalConfigurationInputEndAction**

Global Configuration Input End Action

- NONE
- SWITCH_AND_LOOP_INPUTS

**GlobalConfigurationLowFramerateInputs**

Global Configuration Low Framerate Inputs

- DISABLED
- ENABLED

**GlobalConfigurationOutputLockingMode**

Global Configuration Output Locking Mode

- EPOCH_LOCKING
- PIPELINE_LOCKING

**GlobalConfigurationOutputTimingSource**

Global Configuration Output Timing Source

- INPUT_CLOCK
- SYSTEM_CLOCK
**H264AdaptiveQuantization**

H264 Adaptive Quantization

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

**H264ColorMetadata**

H264 Color Metadata

- IGNORE
- INSERT

**H264EntropyEncoding**

H264 Entropy Encoding

- CABAC
- CAVLC

**H264FlickerAq**

H264 Flicker Aq

- DISABLED
- ENABLED

**H264FramerateControl**

H264 Framerate Control

- INITIALIZE_FROM_SOURCE
- SPECIFIED

**H264GopBReference**

H264 Gop BReference

- DISABLED
- ENABLED

**H264GopSizeUnits**

H264 Gop Size Units

- FRAMES
- SECONDS
**H264Level**

H264 Level

- H264_LEVEL_1
- H264_LEVEL_1_1
- H264_LEVEL_1_2
- H264_LEVEL_1_3
- H264_LEVEL_2
- H264_LEVEL_2_1
- H264_LEVEL_2_2
- H264_LEVEL_3
- H264_LEVEL_3_1
- H264_LEVEL_3_2
- H264_LEVEL_4
- H264_LEVEL_4_1
- H264_LEVEL_4_2
- H264_LEVEL_5
- H264_LEVEL_5_1
- H264_LEVEL_5_2
- H264_LEVEL_AUTO

**H264LookAheadRateControl**

H264 Look Ahead Rate Control

- HIGH
- LOW
- MEDIUM

**H264ParControl**

H264 Par Control

- INITIALIZE_FROM_SOURCE
- SPECIFIED

**H264Profile**

H264 Profile

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

**H264RateControlMode**

H264 Rate Control Mode
CBR
QVBR
VBR

**H264ScanType**

H264 Scan Type

INTERLACED
PROGRESSIVE

**H264SceneChangeDetect**

H264 Scene Change Detect

DISABLED
ENABLED

**H264Settings**

H264 Settings

**minIInterval**

Only meaningful if sceneChangeDetect is set to enabled. 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. Note: Maximum GOP stretch = GOP size + Min-I-interval - 1

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

**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. This field is optional; when no value is specified the encoder will choose the number of slices based on encode resolution.

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

**parNumerator**

Pixel Aspect Ratio numerator.

  * Type: integer
Properties

**Required:** False

**gopSizeUnits**
Indicates if the gopSize is specified in frames or seconds. If seconds the system will convert the gopSize into a frame count at run time.

**Type:** H264GopSizeUnits (p. 486)
**Required:** False

**subgopLength**
If set to fixed, use gopNumBFrames B-frames per sub-GOP. If set to dynamic, optimize the number of B-frames used for each sub-GOP to improve visual quality.

**Type:** H264SubGopLength (p. 494)
**Required:** False

**maxBitrate**
For QVBR: See the tooltip for Quality level For VBR: Set the maximum bitrate in order to accommodate expected spikes in the complexity of the video.

**Type:** integer
**Required:** False
**Minimum:** 1000

**bitrate**
Average bitrate in bits/second. Required when the rate control mode is VBR or CBR. Not used for QVBR. In an MS Smooth output group, each output must have a unique value when its bitrate is rounded down to the nearest multiple of 1000.

**Type:** integer
**Required:** False
**Minimum:** 1000

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

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

**temporalAq**
If set to enabled, adjust quantization within each frame based on temporal variation of content complexity.

**Type:** H264TemporalAq (p. 494)
**Required:** False
**afdSignaling**
Indicates that AFD values will be written into the output stream. If afdSignaling is "auto", the system will try to preserve the input AFD value (in cases where multiple AFD values are valid). If set to "fixed", the AFD value will be the value configured in the fixedAfd parameter.

- **Type**: AfdSignaling (p. 449)
- **Required**: False

**timecodeInsertion**
Determines how timecodes should be inserted into the video elementary stream. - 'disabled': Do not include timecodes - 'picTimingSei': Pass through picture timing SEI messages from the source specified in Timecode Config

- **Type**: H264TimecodeInsertionBehavior (p. 494)
- **Required**: False

**bufSize**
Size of buffer (HRD buffer model) in bits/second.

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

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

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

**framerateControl**
This field indicates how the output video frame rate is specified. If "specified" is selected then the output video frame rate is determined by framerateNumerator and framerateDenominator, else if "initializeFromSource" is selected then the output video frame rate will be set equal to the input video frame rate of the first input.

- **Type**: H264FramerateControl (p. 486)
- **Required**: False

**qvbrQualityLevel**
Controls the target quality for the video encode. Applies only when the rate control mode is QVBR. Set values for the QVBR quality level field and Max bitrate field that suit your most important viewing devices. Recommended values are: - Primary screen: Quality level: 8 to 10. Max bitrate: 4M - PC or tablet: Quality level: 7. Max bitrate: 1.5M to 3M - Smartphone: Quality level: 6. Max bitrate: 1M to 1.5M

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 10
fixedAfd

Four bit AFD value to write on all frames of video in the output stream. Only valid when afdSignaling is set to 'Fixed'.

Type: FixedAfd (p. 483)
Required: False

level

H.264 Level.

Type: H264Level (p. 487)
Required: False

lookAheadRateControl

Amount of lookahead. A value of low can decrease latency and memory usage, while high can produce better quality for certain content.

Type: H264LookAheadRateControl (p. 487)
Required: False

profile

H.264 Profile.

Type: H264Profile (p. 487)
Required: False

framerateNumerator

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

Type: integer
Required: False
Minimum: 1

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

entropyEncoding

Entropy encoding mode. Use cabac (must be in Main or High profile) or cavlc.

Type: H264EntropyEncoding (p. 486)
Required: False
framerateDenominator

Framerate denominator.

Type: integer
Required: False
Minimum: 1

spatialAq

If set to enabled, adjust quantization within each frame based on spatial variation of content complexity.

Type: H264SpatialAq (p. 494)
Required: False

adaptiveQuantization

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

Type: H264AdaptiveQuantization (p. 486)
Required: False

colorMetadata

Includes colorspace metadata in the output.

Type: H264ColorMetadata (p. 486)
Required: False

gopSize

GOP size (keyframe interval) in units of either frames or seconds per gopSizeUnits. Must be greater than zero.

Type: number
Required: False
Minimum: 1.0

numRefFrames

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

gopBReference

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

Type: H264GopBReference (p. 486)
Required: False
**parControl**

This field indicates how the output pixel aspect ratio is specified. If "specified" is selected then the output video pixel aspect ratio is determined by parNumerator and parDenominator, else if "initializeFromSource" is selected then the output pixel aspect ratio will be set equal to the input video pixel aspect ratio of the first input.

- **Type**: H264ParControl (p. 487)
- **Required**: False

**parDenominator**

Pixel Aspect Ratio denominator.

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

**syntax**

Produces a bitstream compliant with SMPTE RP-2027.

- **Type**: H264Syntax (p. 494)
- **Required**: False

**sceneChangeDetect**

Scene change detection. - On: inserts I-frames when scene change is detected. - Off: does not force an I-frame when scene change is detected.

- **Type**: H264SceneChangeDetect (p. 488)
- **Required**: False

**scanType**

Sets the scan type of the output to progressive or top-field-first interlaced.

- **Type**: H264ScanType (p. 488)
- **Required**: False

**flickerAq**

If set to enabled, adjust quantization within each frame to reduce flicker or ‘pop’ on I-frames.

- **Type**: H264FlickerAq (p. 486)
- **Required**: False

**gopNumBFrames**

Number of B-frames between reference frames.

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

Rate control mode. QVBR: Quality will match the specified quality level except when it is constrained by the maximum bitrate. Recommended if you or your viewers pay for bandwidth. VBR: Quality and bitrate vary, depending on the video complexity. Recommended instead of QVBR if you want to maintain a specific average bitrate over the duration of the channel. CBR: Quality varies, depending on the video complexity. Recommended only if you distribute your assets to devices that cannot handle variable bitrates.

Type: H264RateControlMode (p. 487)
Required: False

H264SpatialAq

H264 Spatial Aq

DISABLED
ENABLED

H264SubGopLength

H264 Sub Gop Length

DYNAMIC
FIXED

H264Syntax

H264 Syntax

DEFAULT
RP2027

H264TemporalAq

H264 Temporal Aq

DISABLED
ENABLED

H264TimecodeInsertionBehavior

H264 Timecode Insertion Behavior

DISABLED
PIC_TIMING_SEI

HlsAdMarkers

Hls Ad Markers

ADOBE
**HlsAkamaiHttpTransferMode**

Hls Akamai Http Transfer Mode

- **CHUNKED**
- **NON_CHUNKED**

**HlsAkamaiSettings**

Hls Akamai Settings

**salt**

Salt for authenticated Akamai.

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

**httpTransferMode**

Specify whether or not to use chunked transfer encoding to Akamai. User should contact Akamai to enable this feature.

- **Type:** HlsAkamaiHttpTransferMode (p. 495)
- **Required:** False

**numRetries**

Number of retry attempts that will be made before the Live Event is put into an error state.

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

**restartDelay**

If a streaming output fails, number of seconds to wait until a restart is initiated. A value of 0 means never restart.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
- **Maximum:** 15

**connectionRetryInterval**

Number of seconds to wait before retrying connection to the CDN if the connection is lost.

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

**filecacheDuration**
Size in seconds of file cache for streaming outputs.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
- **Maximum:** 600

**token**
Token parameter for authenticated akamai. If not specified, _gda_ is used.

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

**HlsBasicPutSettings**
Hls Basic Put Settings

**numRetries**
Number of retry attempts that will be made before the Live Event is put into an error state.

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

**restartDelay**
If a streaming output fails, number of seconds to wait until a restart is initiated. A value of 0 means never restart.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
- **Maximum:** 15

**connectionRetryInterval**
Number of seconds to wait before retrying connection to the CDN if the connection is lost.

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

**filecacheDuration**
Size in seconds of file cache for streaming outputs.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
Maximum: 600

**HlsCaptionLanguageSetting**

Hls Caption Language Setting

- INSERT
- NONE
- OMIT

**HlsCdnSettings**

Hls Cdn Settings

**hlsAkamaiSettings**

Type: HlsAkamaiSettings (p. 495)
Required: False

**hlsWebdavSettings**

Type: HlsWebdavSettings (p. 508)
Required: False

**hlsBasicPutSettings**

Type: HlsBasicPutSettings (p. 496)
Required: False

**hlsMediaStoreSettings**

Type: HlsMediaStoreSettings (p. 505)
Required: False

**HlsClientCache**

Hls Client Cache

- DISABLED
- ENABLED

**HlsCodecSpecification**

Hls Codec Specification

- RFC_4281
- RFC_6381

**HlsDirectoryStructure**

Hls Directory Structure
SINGLE_DIRECTORY
SUBDIRECTORY_PER_STREAM

HlsEncryptionType

Hls Encryption Type

AES128
SAMPLE_AES

HlsGroupSettings

Hls Group Settings

segmentsPerSubdirectory

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

Type: integer
Required: False
Minimum: 1

ivInManifest

For use with encryptionType. The IV (Initialization Vector) is a 128-bit number used in conjunction with the key for encrypting blocks. If set to "include", IV is listed in the manifest, otherwise the IV is not in the manifest.

Type: HlsIvInManifest (p. 504)
Required: False

outputSelection

MANIFESTSANDSEGMENTS: Generates manifests (master manifest, if applicable, and media manifests) for this output group. SEGMENTSONLY: Does not generate any manifests for this output group.

Type: HlsOutputSelection (p. 506)
Required: False

destination

A directory or HTTP destination for the HLS segments, manifest files, and encryption keys (if enabled).

Type: OutputLocationRef (p. 536)
Required: True

encryptionType

Encrypts the segments with the given encryption scheme. Exclude this parameter if no encryption is desired.

Type: HlsEncryptionType (p. 498)
Required: False

**indexNSegments**
Applies only if Mode field is LIVE. Specifies the maximum number of segments in the media manifest file. After this maximum, older segments are removed from the media manifest. This number must be less than or equal to the Keep Segments field.

Type: integer
  Required: False
  Minimum: 3

**constantIv**
For use with encryptionType. This is a 128-bit, 16-byte hex value represented by a 32-character text string. If ivSource is set to "explicit" then this parameter is required and is used as the IV for encryption.

Type: string
  Required: False
  MinLength: 32
  MaxLength: 32

**timedMetadataId3Frame**
Indicates ID3 frame that has the timecode.

Type: HlsTimedMetadataId3Frame (p. 507)
  Required: False

**baseUrlManifest**
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

**captionLanguageSetting**
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.

Type: HlsCaptionLanguageSetting (p. 497)
  Required: False

**minSegmentLength**
When set, minimumSegmentLength 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: False
Minimum: 0

mode

If "vod", all segments are indexed and kept permanently in the destination and manifest. If "live", only
the number segments specified in keepSegments and indexNSegments are kept; newer segments replace
older segments, which may prevent players from rewinding all the way to the beginning of the event.
VOD mode uses HLS EXT-X-PLAYLIST-TYPE of EVENT while the channel is running, converting it to a
"VOD" type manifest on completion of the stream.

Type: HlsMode (p. 506)
Required: False

keyProviderSettings

The key provider settings.

Type: KeyProviderSettings (p. 515)
Required: False

manifestCompression

When set to gzip, compresses HLS playlist.

Type: HlsManifestCompression (p. 505)
Required: False

ivSource

For use with encryptionType. The IV (Initialization Vector) is a 128-bit number used in conjunction with
the key for encrypting blocks. If this setting is "followsSegmentNumber", it will cause the IV to change
every segment (to match the segment number). If this is set to "explicit", you must enter a constantIV
value.

Type: HlsIvSource (p. 504)
Required: False

tsFileMode

SEGMENTEDFILES: Emit the program as segments - multiple .ts media files. SINGLEFILE: Applies only
if Mode field is VOD. Emit the program as a single .ts media file. The media manifest includes #EXT-X-
BYTERANGE tags to index segments for playback. A typical use for this value is when sending the output
to AWS Elemental MediaConvert, which can accept only a single media file. Playback while the channel is
running is not guaranteed due to HTTP server caching.

Type: HlsTsFileMode (p. 508)
Required: False

manifestDurationFormat

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

Type: HlsManifestDurationFormat (p. 505)
Required: False

**keyFormatVersions**

Either a single positive integer version value or a slash delimited list of version values (1/2/3).

Type: string

Required: False

**streamInfResolution**

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

Type: HlsStreamInfResolution (p. 507)

Required: False

**timestampDeltaMilliseconds**

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

Type: integer

Required: False

Minimum: 0

**baseUrlContent**

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

**segmentationMode**

useInputSegmentation has been deprecated. The configured segment size is always used.

Type: HlsSegmentationMode (p. 507)

Required: False

**captionLanguageMappings**

Mapping of up to 4 caption channels to caption languages. Is only meaningful if captionLanguageSetting is set to "insert".

Type: Array of type CaptionLanguageMapping (p. 464)

Required: False

**clientCache**

When set to "disabled", sets the #EXT-X-ALLOW-CACHE:no tag in the manifest, which prevents clients from saving media segments for later replay.

Type: HlsClientCache (p. 497)
Properties

Required: False

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

  Type: HlsCodecSpecification (p. 497)
  Required: False

keepSegments
Applies only if Mode field is LIVE. Specifies the number of media segments (.ts files) to retain in the destination directory.

  Type: integer
  Required: False
  Minimum: 1

redundantManifest
ENABLED: The master manifest (.m3u8 file) for each pipeline includes information about both pipelines: first its own media files, then the media files of the other pipeline. This feature allows playout device that support stale manifest detection to switch from one manifest to the other, when the current manifest seems to be stale. There are still two destinations and two master manifests, but both master manifests reference the media files from both pipelines. DISABLED: The master manifest (.m3u8 file) for each pipeline includes information about its own pipeline only. For an HLS output group with MediaPackage as the destination, the DISABLED behavior is always followed. MediaPackage regenerates the manifests it serves to players so a redundant manifest from MediaLive is irrelevant.

  Type: HlsRedundantManifest (p. 507)
  Required: False

timedMetadataId3Period
Timed Metadata interval in seconds.

  Type: integer
  Required: False
  Minimum: 0

programDateTime
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 timestampOffset.

  Type: HlsProgramDateTime (p. 507)
  Required: False

directoryStructure
Place segments in subdirectories.

  Type: HlsDirectoryStructure (p. 497)
  Required: False
keyFormat

The value specifies how the key is represented in the resource identified by the URI. If parameter is absent, an implicit value of "identity" is used. A reverse DNS string can also be given.

Type: string
Required: False

inputLossAction

Parameter that control output group behavior on input loss.

Type: InputLossActionForHlsOut (p. 511)
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 HlsAdMarkers (p. 494)
Required: False

programDateTimePeriod

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

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

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: False
Minimum: 1

hlsCdnSettings

Parameters that control interactions with the CDN.

Type: HlsCdnSettings (p. 497)
Required: False

iFrameOnlyPlaylists

DISABLED: Do not create an I-frame-only manifest, but do create the master and media manifests (according to the Output Selection field). STANDARD: Create an I-frame-only manifest for each output that contains video, as well as the other manifests (according to the Output Selection field). The I-frame manifest contains a #EXT-X-I-FRAMES-ONLY tag to indicate it is I-frame only, and one or more #EXT-X-BYTERANGE entries identifying the I-frame position. For example, #EXT-X-BYTERANGE:160364@1461888"
**Properties**

**Type**: `IFrameOnlyPlaylistType` (p. 509)
**Required**: False

**HlsInputSettings**

Hls Input Settings

**retries**

The number of consecutive times that attempts to read a manifest or segment must fail before the input is considered unavailable.

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

**bandwidth**

When specified the HLS stream with the m3u8 BANDWIDTH that most closely matches this value will be chosen, otherwise the highest bandwidth stream in the m3u8 will be chosen. The bitrate is specified in bits per second, as in an HLS manifest.

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

**retryInterval**

The number of seconds between retries when an attempt to read a manifest or segment fails.

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

**bufferSegments**

When specified, reading of the HLS input will begin this many buffer segments from the end (most recently written segment). When not specified, the HLS input will begin with the first segment specified in the m3u8.

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

**HlsIvInManifest**

Hls Iv In Manifest

EXCLUDE
INCLUDE

**HlsIvSource**

Hls Iv Source
EXPLICIT
FOLLOWS_SEGMENT_NUMBER

**HlsManifestCompression**

Hls Manifest Compression

- GZIP
- NONE

**HlsManifestDurationFormat**

Hls Manifest Duration Format

- FLOATING_POINT
- INTEGER

**HlsMediaStoreSettings**

Hls Media Store Settings

**mediaStoreStorageClass**

When set to temporal, output files are stored in non-persistent memory for faster reading and writing.

- **Type:** HlsMediaStoreStorageClass (p. 506)
- **Required:** False

**numRetries**

Number of retry attempts that will be made before the Live Event is put into an error state.

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

**restartDelay**

If a streaming output fails, number of seconds to wait until a restart is initiated. A value of 0 means never restart.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
- **Maximum:** 15

**connectionRetryInterval**

Number of seconds to wait before retrying connection to the CDN if the connection is lost.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
Properties

**filecacheDuration**

Size in seconds of file cache for streaming outputs.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 600

**HlsMediaStoreStorageClass**

Hls Media Store Storage Class

- TEMPORAL

**HlsMode**

Hls Mode

- LIVE
- VOD

**HlsOutputSelection**

Hls Output Selection

- MANIFESTS_AND_SEGMENTS
- SEGMENTS_ONLY

**HlsOutputSettings**

Hls Output Settings

**segmentModifier**

String concatenated to end of segment filenames.

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

**hlsSettings**

Settings regarding the underlying stream. These settings are different for audio-only outputs.

- **Type**: HlsSettings (p. 507)
- **Required**: True

**nameModifier**

String concatenated to the end of the destination filename. Accepts `"Format Identifiers \:\#formatIdentifierParameters`.

- **Type**: string
- **Required**: False
MinLength: 1

**HlsProgramDateTime**
Hls Program Date Time

EXCLUDE
INCLUDE

**HlsRedundantManifest**
Hls Redundant Manifest

DISABLED
ENABLED

**HlsSegmentationMode**
Hls Segmentation Mode

USE_INPUT_SEGMENTATION
USE_SEGMENT_DURATION

**HlsSettings**
Hls Settings

*standardHlsSettings*

Type: StandardHlsSettings (p. 544)
Required: False

*audioOnlyHlsSettings*

Type: AudioOnlyHlsSettings (p. 454)
Required: False

**HlsStreamInfResolution**
Hls Stream Inf Resolution

EXCLUDE
INCLUDE

**HlsTimedMetadataId3Frame**
Hls Timed Metadata Id3 Frame

NONE
PRIV
TDRL
HlsTsFileMode

Hls Ts File Mode

- SEGMENTED_FILES
- SINGLE_FILE

HlsWebdavHttpTransferMode

Hls Webdav Http Transfer Mode

- CHUNKED
- NON_CHUNKED

HlsWebdavSettings

Hls Webdav Settings

httpTransferMode

Specify whether or not to use chunked transfer encoding to WebDAV.

- Type: HlsWebdavHttpTransferMode (p. 508)
- Required: False

numRetries

Number of retry attempts that will be made before the Live Event is put into an error state.

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

restartDelay

If a streaming output fails, number of seconds to wait until a restart is initiated. A value of 0 means never restart.

- Type: integer
- Required: False
- Minimum: 0
- Maximum: 15

connectionRetryInterval

Number of seconds to wait before retrying connection to the CDN if the connection is lost.

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

filecacheDuration

Size in seconds of file cache for streaming outputs.
Properties

**Type**: integer
**Required**: False
**Minimum**: 0
**Maximum**: 600

**IframeOnlyPlaylistType**

When set to “standard”, an I-Frame only playlist will be written out for each video output in the output group. This I-Frame only playlist will contain byte range offsets pointing to the I-frame(s) in each segment.

- DISABLED
- STANDARD

**InputAttachment**

**inputId**

The ID of the input

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

**inputAttachmentName**

User-specified name for the attachment. This is required if the user wants to use this input in an input switch action.

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

**inputSettings**

Settings of an input (caption selector, etc.)

- **Type**: `InputSettings (p. 513)`
- **Required**: False

**InputChannelLevel**

Input Channel Level

**inputChannel**

The index of the input channel used as a source.

- **Type**: integer
- **Required**: True
- **Minimum**: 0
- **Maximum**: 15

**gain**

Remixing value. Units are in dB and acceptable values are within the range from -60 (mute) and 6 dB.
**Type**: integer  
**Required**: True  
**Minimum**: -60  
**Maximum**: 6

**InputCodec**

Codec in increasing order of complexity

- MPEG2
- AVC
- HEVC

**InputDeblockFilter**

Input Deblock Filter

- DISABLED
- ENABLED

**InputDenoiseFilter**

Input Denoise Filter

- DISABLED
- ENABLED

**InputFilter**

Input Filter

- AUTO
- DISABLED
- FORCED

**InputLocation**

Input Location

**passwordParam**

Key used to extract the password from EC2 Parameter store

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

**uri**

Uniform Resource Identifier - This should be a path to a file accessible to the Live system (eg. a http:// URI) depending on the output type. For example, a RTMP destination should have a uri similar to: “rtmp://fmsserver/live”.

510
Type: string
Required: True

**username**

Username if credentials are required to access a file or publishing point. This can be either a plaintext username, or a reference to an AWS parameter store name from which the username can be retrieved. AWS Parameter store format: "ssm://<parameter name>"

Type: string
Required: False

**InputLossActionForHlsOut**

Input Loss Action For Hls Out

- EMIT_OUTPUT
- PAUSE_OUTPUT

**InputLossActionForMsSmoothOut**

Input Loss Action For Ms Smooth Out

- EMIT_OUTPUT
- PAUSE_OUTPUT

**InputLossActionForRtmpOut**

Input Loss Action For Rtmp Out

- EMIT_OUTPUT
- PAUSE_OUTPUT

**InputLossActionForUdpOut**

Input Loss Action For Udp Out

- DROP_PROGRAM
- DROP_TS
- EMIT_PROGRAM

**InputLossBehavior**

Input Loss Behavior

**inputLossImageType**

Indicates whether to substitute a solid color or a slate into the output after input loss exceeds blackFrameMsec.

Type: InputLossImageType (p. 512)
Properties

**Required**: False

**inputLossImageColor**

When input loss image type is "color" this field specifies the color to use. Value: 6 hex characters representing the values of RGB.

- **Type**: string
- **Required**: False
- **MinLength**: 6
- **MaxLength**: 6

**inputLossImageSlate**

When input loss image type is "slate" these fields specify the parameters for accessing the slate.

- **Type**: InputLocation (p. 510)
- **Required**: False

**repeatFrameMsec**

On input loss, the number of milliseconds to repeat the previous picture before substituting black into the output. A value x, where 0 ≤ x ≤ 1,000,000 and a value of 1,000,000 will be interpreted as infinite.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 1000000

**blackFrameMsec**

On input loss, the number of milliseconds to substitute black into the output before switching to the frame specified by inputLossImageType. A value x, where 0 ≤ x ≤ 1,000,000 and a value of 1,000,000 will be interpreted as infinite.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 1000000

**InputLossImageType**

Input Loss Image Type

- COLOR
- SLATE

**InputMaximumBitrate**

Maximum input bitrate in megabits per second. Bitrates up to 50 Mbps are supported currently.

- MAX_10_MBPS
- MAX_20_MBPS
- MAX_50_MBPS
InputResolution
Input resolution based on lines of vertical resolution in the input; SD is less than 720 lines, HD is 720 to 1080 lines, UHD is greater than 1080 lines
- SD
- HD
- UHD

InputSettings
Live Event input parameters. There can be multiple inputs in a single Live Event.

sourceEndBehavior
Loop input if it is a file. This allows a file input to be streamed indefinitely.

  Type: InputSourceEndBehavior (p. 514)
  Required: False

deblockFilter
Enable or disable the deblock filter when filtering.

  Type: InputDeblockFilter (p. 510)
  Required: False

audioSelectors
Used to select the audio stream to decode for inputs that have multiple available.

  Type: Array of type AudioSelector (p. 455)
  Required: False

networkInputSettings
Input settings.

  Type: NetworkInputSettings (p. 533)
  Required: False

inputFilter
Turns on the filter for this input. MPEG-2 inputs have the deblocking filter enabled by default. 1) auto - filtering will be applied depending on input type/quality 2) disabled - no filtering will be applied to the input 3) forced - filtering will be applied regardless of input type

  Type: InputFilter (p. 510)
  Required: False

videoSelector
Informs which video elementary stream to decode for input types that have multiple available.

  Type: VideoSelector (p. 549)
Properties

**filterStrength**
Adjusted to the magnitude of filtering from 1 (minimal) to 5 (strongest).

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 5

**denoiseFilter**
Enable or disable the denoise filter when filtering.

- **Type**: InputDenoiseFilter (p. 510)
- **Required**: False

**captionSelectors**
Used to select the caption input to use for inputs that have multiple available.

- **Type**: Array of type CaptionSelector (p. 465)
- **Required**: False

**InputSourceEndBehavior**
Input Source End Behavior

- CONTINUE
- LOOP

**InputSpecification**

**codec**
Input codec

- **Type**: InputCodec (p. 510)
- **Required**: False

**resolution**
Input resolution, categorized coarsely

- **Type**: InputResolution (p. 513)
- **Required**: False

**maximumBitrate**
Maximum input bitrate, categorized coarsely

- **Type**: InputMaximumBitrate (p. 512)
- **Required**: False
**InternalServiceError**

*message*

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

**InvalidRequest**

*message*

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

**KeyProviderSettings**

Key Provider Settings

*staticKeySettings*

- **Type:** StaticKeySettings (p. 544)
- **Required:** False

**LimitExceeded**

*message*

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

**LogLevel**

The log level the user wants for their channel.

- ERROR
- WARNING
- INFO
- DEBUG
- DISABLED

**M2tsAbsentInputAudioBehavior**

M2ts Absent Input Audio Behavior

- DROP
- ENCODE_SILENCE

**M2tsArib**

M2ts Arib
DISABLED
ENABLED

**M2tsAribCaptionsPidControl**

M2ts Arib Captions Pid Control

AUTO
USE_CONFIGURED

**M2tsAudioBufferModel**

M2ts Audio Buffer Model

ATSC
DVB

**M2tsAudioInterval**

M2ts Audio Interval

VIDEO_AND_FIXED_INTERVALS
VIDEO_INTERVAL

**M2tsAudioStreamType**

M2ts Audio Stream Type

ATSC
DVB

**M2tsBufferModel**

M2ts Buffer Model

MULTIPLEX
NONE

**M2tsCcDescriptor**

M2ts Cc Descriptor

DISABLED
ENABLED

**M2tsEbifControl**

M2ts Ebif Control

NONE
PASSTHROUGH

**M2tsEbpPlacement**

M2ts Ebp Placement

- VIDEO_AND_AUDIO_PIDS
- VIDEO_PID

**M2tsEsRateInPes**

M2ts Es Rate In PEs

- EXCLUDE
- INCLUDE

**M2tsKlv**

M2ts Klv

- NONE
- PASSTHROUGH

**M2tsPcrControl**

M2ts Pcr Control

- CONFIGURED_PCR_PERIOD
- PCR_EVERY_PES_PACKET

**M2tsRateMode**

M2ts Rate Mode

- CBR
- VBR

**M2tsScte35Control**

M2ts Scte35 Control

- NONE
- PASSTHROUGH

**M2tsSegmentationMarkers**

M2ts Segmentation Markers

- EBP
- EBP_LEGACY
NONE
PSI_SEGSTART
RAI_ADAPT
RAI_SEGSTART

**M2tsSegmentationStyle**

M2ts Segmentation Style

- MAINTAIN_CADENCE
- RESET_CADENCE

**M2tsSettings**

M2ts Settings

**audioStreamType**

When set to atsc, uses stream type = 0x81 for AC3 and stream type = 0x87 for EAC3. When set to dvb, uses stream type = 0x06.

*Type:* M2tsAudioStreamType (p. 516)
*Required:* False

**ecmPid**

This field is unused and deprecated.

*Type:* string
*Required:* False

**dvbTeletextPid**

Packet Identifier (PID) for input source DVB Teletext data to this output. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

*Type:* string
*Required:* False

**aribCaptionsPidControl**

If set to auto, pid number used for ARIB Captions will be auto-selected from unused pids. If set to useConfigured, ARIB Captions will be on the configured pid number.

*Type:* M2tsAribCaptionsPidControl (p. 516)
*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.

*Type:* integer
*Required:* False
Properties

Minimum: 0

rateMode

When vbr, does not insert null packets into transport stream to fill specified bitrate. The bitrate setting acts as the maximum bitrate when vbr is set.

Type: M2tsRateMode (p. 517)
Required: False

segmentationTime

The length in seconds of each segment. Required unless markers is set to None.

Type: number
Required: False
Minimum: 1.0

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. Can be entered as decimal or hexadecimal values. Each PID specified must be in the range of 32 (or 0x20)..8182 (or 0x1ff6).

Type: string
Required: False

audioFramesPerPes

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

Type: integer
Required: False
Minimum: 0

fragmentTime

The length in seconds of each fragment. Only used with EBP markers.

Type: number
Required: False
Minimum: 0.0

ebpLookaheadMs

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

### ebpAudioInterval

When `videoAndFixedIntervals` is selected, 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. Only available when EBP Cablelabs segmentation markers are selected. Partitions 1 and 2 will always follow the video interval.

*Type: M2tsAudioInterval (p. 516)*

*Required: False*

### scte35Pid

Packet Identifier (PID) of the SCTE-35 stream in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

*Type: string*

*Required: False*

### programNum

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

*Type: integer*

*Required: False*

*Minimum: 0*

*Maximum: 65535*

### pmtInterval

The number of milliseconds between instances of this table in the output transport stream. Valid values are 0, 10..1000.

*Type: integer*

*Required: False*

*Minimum: 0*

*Maximum: 1000*

### pcrPeriod

Maximum time in milliseconds between Program Clock Reference (PCRs) inserted into the transport stream.

*Type: integer*

*Required: False*

*Minimum: 0*

*Maximum: 500*

### segmentationStyle

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 "resetCadence" 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 $segmentationTime seconds. When a segmentation style of "maintainCadence" 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 $segmentationTime seconds. Note that EBP lookahead is a slight exception to this rule.

Type: $M2tsSegmentationStyle (p. 518)
Required: False

ebif

If set to passthrough, passes any EBIF data from the input source to this output.

Type: $M2tsEbifControl (p. 516)
Required: False

audioBufferModel

When set to dvb, uses DVB buffer model for Dolby Digital audio. When set to atsc, the ATSC model is used.

Type: $M2tsAudioBufferModel (p. 516)
Required: False

dvbNitSettings

Inserts DVB Network Information Table (NIT) at the specified table repetition interval.

Type: $DvbNitSettings (p. 469)
Required: False

absentInputAudioBehavior

When set to drop, output audio streams will be removed from the program if the selected input audio stream is removed from the input. This allows the output audio configuration to dynamically change based on input configuration. If this is set to encodeSilence, all output audio streams will output encoded silence when not connected to an active input stream.

Type: $M2tsAbsentInputAudioBehavior (p. 515)
Required: False

timedMetadataBehavior

When set to passthrough, timed metadata will be passed through from input to output.

Type: $M2tsTimedMetadataBehavior (p. 525)
Required: False

timedMetadataPid

Packet Identifier (PID) of the timed metadata stream in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

Type: string
Required: False
### pmtPid
Packet Identifier (PID) for the Program Map Table (PMT) in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20).8182 (or 0x1ff6).

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

### etvSignalPid
Packet Identifier (PID) for input source ETV Signal data to this output. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20).8182 (or 0x1ff6).

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

### bufferModel
If set to multiplex, use multiplex buffer model for accurate interleaving. Setting to bufferModel to none can lead to lower latency, but low-memory devices may not be able to play back the stream without interruptions.

**Type:** M2tsBufferModel (p. 516)  
**Required:** False

### scte35Control
Optionally pass SCTE-35 signals from the input source to this output.

**Type:** M2tsScte35Control (p. 517)  
**Required:** False

### ebpPlacement
Controls placement of EBP on Audio PIDs. If set to videoAndAudioPids, EBP markers will be placed on the video PID and all audio PIDs. If set to videoPid, EBP markers will be placed on only the video PID.

**Type:** M2tsEbpPlacement (p. 517)  
**Required:** False

### arib
When set to enabled, uses ARIB-compliant field muxing and removes video descriptor.

**Type:** M2tsArib (p. 515)  
**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
Minimum: 0.0

dvbSdtSettings
Inserts DVB Service Description Table (SDT) at the specified table repetition interval.

Type: DvbSdtSettings (p. 469)  
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. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20).8182 (or 0x1ff6).

Type: string  
Required: False

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

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

pcrControl
When set to pcrEveryPesPacket, 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.

Type: M2tsPcrControl (p. 517)  
Required: False

videoPid
Packet Identifier (PID) of the elementary video stream in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20).8182 (or 0x1ff6).

Type: string  
Required: False

esRateInPes
Include or exclude the ES Rate field in the PES header.

Type: M2tsEsRateInPes (p. 517)  
Required: False

segmentationMarkers
Inserts segmentation markers at each segmentationTime period. raiSegstart sets the Random Access Indicator bit in the adaptation field. raiAdapt sets the RAI bit and adds the current timecode in the
private data bytes. psiSegstart 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. ebpLegacy adds Encoder Boundary Point information to the adaptation field using a legacy proprietary format.

**Type**: M2tsSegmentationMarkers (p. 517)

**Required**: False

### dvbTdtSettings

Inserts DVB Time and Date Table (TDT) at the specified table repetition interval.

**Type**: DvbTdtSettings (p. 474)

**Required**: False

### klv

If set to passthrough, passes any KLV data from the input source to this output.

**Type**: M2tsKlv (p. 517)

**Required**: False

### ccDescriptor

When set to enabled, generates captionServiceDescriptor in PMT.

**Type**: M2tsCcDescriptor (p. 516)

**Required**: False

### patInterval

The number of milliseconds between instances of this table in the output transport stream. Valid values are 0, 10..1000.

**Type**: integer

**Required**: False

**Minimum**: 0

**Maximum**: 1000

### etvPlatformPid

Packet Identifier (PID) for input source ETV Platform data to this output. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

**Type**: string

**Required**: False

### 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. Can be entered as decimal or hexadecimal values. Each PID specified must be in the range of 32 (or 0x20)..8182 (or 0x1ff6).

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

**aribCaptionsPid**
Packet Identifier (PID) for ARIB Captions in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

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

**scte27Pids**
Packet Identifier (PID) for input source SCTE-27 data to this output. Multiple values are accepted, and can be entered in ranges and/or by comma separation. Can be entered as decimal or hexadecimal values. Each PID specified must be in the range of 32 (or 0x20)..8182 (or 0x1ff6).

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

**klvDataPids**
Packet Identifier (PID) for input source KLV data to this output. Multiple values are accepted, and can be entered in ranges and/or by comma separation. Can be entered as decimal or hexadecimal values. Each PID specified must be in the range of 32 (or 0x20)..8182 (or 0x1ff6).

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

**M2tsTimedMetadataBehavior**
M2ts Timed Metadata Behavior

- NO_PASSTHROUGH
- PASSTHROUGH

**M3u8PcrControl**
M3u8 Pcr Control

- CONFIGURED_PCR_PERIOD
- PCR_EVERY_PES_PACKET

**M3u8Scte35Behavior**
M3u8 Scte35 Behavior

- NO_PASSTHROUGH
- PASSTHROUGH

**M3u8Settings**
Settings information for the .m3u8 container
pmtPid
Packet Identifier (PID) for the Program Map Table (PMT) in the transport stream. Can be entered as a decimal or hexadecimal value.

Type: string
Required: False

ecmPid
This parameter is unused and deprecated.

Type: string
Required: False

scte35Behavior
If set to passthrough, passes any SCTE-35 signals from the input source to this output.

Type: M3u8Scte35Behavior (p. 525)
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. Can be entered as a decimal or hexadecimal value.

Type: string
Required: False

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. Can be entered as decimal or hexadecimal values.

Type: string
Required: False

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

Type: integer
Required: False
Minimum: 0

scte35Pid
Packet Identifier (PID) of the SCTE-35 stream in the transport stream. Can be entered as a decimal or hexadecimal value.

Type: string
Properties

transportStreamId

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

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

pcrControl

When set to pcrEveryPesPacket, 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.

Type: M3u8PcrControl (p. 525)
Required: False

videoPid

Packet Identifier (PID) of the elementary video stream in the transport stream. Can be entered as a decimal or hexadecimal value.

Type: string
Required: False

pcrPeriod

Maximum time in milliseconds between Program Clock References (PCRs) inserted into the transport stream.

Type: integer
Required: False
Minimum: 0
Maximum: 500

pmtInterval

The number of milliseconds between instances of this table in the output transport stream. A value of "0" writes out the PMT once per segment file.

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

programNum

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

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

**patInterval**

The number of milliseconds between instances of this table in the output transport stream. A value of "0" writes out the PMT once per segment file.

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

**timedMetadataPid**

Packet Identifier (PID) of the timed metadata stream in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

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

**timedMetadataBehavior**

When set to passthrough, timed metadata is passed through from input to output.

- **Type**: M3u8TimedMetadataBehavior (p. 528)
- **Required**: False

**M3u8TimedMetadataBehavior**

M3u8 Timed Metadata Behavior

- NO_PASSTHROUGH
- PASSTHROUGH

**MediaPackageGroupSettings**

Media Package Group Settings

**destination**

MediaPackage channel destination.

- **Type**: OutputLocationRef (p. 536)
- **Required**: True

**MediaPackageOutputDestinationSettings**

Media Package Output Destination Settings

**channelId**

ID of the channel in MediaPackage that is the destination for this output group. You do not need to specify the individual inputs in MediaPackage; MediaLive will handle the connection of the two
MediaLive pipelines to the two MediaPackage inputs. The MediaPackage channel and MediaLive channel must be in the same region.

- **Type**: string
  - **Required**: False
  - **MinLength**: 1

### MediaPackageOutputSettings

Media Package Output Settings

#### Mp2CodingMode

Mp2 Coding Mode

- CODING_MODE_1_0
- CODING_MODE_2_0

#### Mp2Settings

Mp2 Settings

- **codingMode**
  - The MPEG2 Audio coding mode. Valid values are codingMode10 (for mono) or codingMode20 (for stereo).
  - **Type**: Mp2CodingMode (p. 529)
  - **Required**: False

- **bitrate**
  - Average bitrate in bits/second.
  - **Type**: number
  - **Required**: False

- **sampleRate**
  - Sample rate in Hz.
  - **Type**: number
  - **Required**: False

### MsSmoothGroupSettings

Ms Smooth Group Settings

- **fragmentLength**
  - Length of mp4 fragments to generate (in seconds). Fragment length must be compatible with GOP size and framerate.
  - **Type**: integer
**eventId**
MS Smooth event ID to be sent to the IIS server. Should only be specified if eventIdMode is set to useConfigured.

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

**timestampOffset**
Timestamp offset for the event. Only used if timestampOffsetMode is set to useConfiguredOffset.

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

**segmentationMode**
useInputSegmentation has been deprecated. The configured segment size is always used.

- **Type**: SmoothGroupSegmentationMode (p. 543)
- **Required**: False

**numRetries**
Number of retry attempts.

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

**eventStopBehavior**
When set to sendEos, send EOS signal to IIS server when stopping the event

- **Type**: SmoothGroupEventStopBehavior (p. 543)
- **Required**: False

**acquisitionPointId**
The value of the "Acquisition Point Identity" element used in each message placed in the sparse track. Only enabled if sparseTrackType is not "none".

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

**sparseTrackType**
If set to scte35, use incoming SCTE-35 messages to generate a sparse track in this group of MS-Smooth outputs.

- **Type**: SmoothGroupSparseTrackType (p. 543)
- **Required**: False
timestampOffsetMode

Type of timestamp date offset to use. - useEventStartDate: Use the date the event was started as the offset - useConfiguredOffset: Use an explicitly configured date as the offset

Type: SmoothGroupTimestampOffsetMode (p. 544)
Required: False

destination

Smooth Streaming publish point on an IIS server. Elemental Live acts as a "Push" encoder to IIS.

Type: OutputLocationRef (p. 536)
Required: True

audioOnlyTimecodeControl

If set to passthrough for an audio-only MS Smooth output, the fragment absolute time will be set to the current timecode. This option does not write timecodes to the audio elementary stream.

Type: SmoothGroupAudioOnlyTimecodeControl (p. 543)
Required: False

connectionRetryInterval

Number of seconds to wait before retrying connection to the IIS server if the connection is lost. Content will be cached during this time and the cache will be delivered to the IIS server once the connection is re-established.

Type: integer
Required: False
Minimum: 0

filecacheDuration

Size in seconds of file cache for streaming outputs.

Type: integer
Required: False
Minimum: 0

certificateMode

If set to verifyAuthenticity, verify the https certificate chain to a trusted Certificate Authority (CA). This will cause https outputs to self-signed certificates to fail.

Type: SmoothGroupCertificateMode (p. 543)
Required: False

inputLossAction

Parameter that control output group behavior on input loss.

Type: InputLossActionForMsSmoothOut (p. 511)
**Required**: False

**sendDelayMs**

Number of milliseconds to delay the output from the second pipeline.

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

**eventIdMode**

Specifies whether or not to send an event ID to the IIS server. If no event ID is sent and the same Live Event is used without changing the publishing point, clients might see cached video from the previous run. Options: - "useConfigured" - use the value provided in eventId - "useTimestamp" - generate and send an event ID based on the current timestamp - "noEventId" - do not send an event ID to the IIS server.

- **Type**: SmoothGroupEventIdMode (p. 543)
- **Required**: False

**restartDelay**

Number of seconds before initiating a restart due to output failure, due to exhausting the numRetries on one segment, or exceeding filecacheDuration.

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

**streamManifestBehavior**

When set to send, send stream manifest so publishing point doesn't start until all streams start.

- **Type**: SmoothGroupStreamManifestBehavior (p. 543)
- **Required**: False

**MsSmoothOutputSettings**

Ms Smooth Output Settings

**nameModifier**

String concatenated to the end of the destination filename. Required for multiple outputs of the same type.

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

**NetworkInputServerValidation**

Network Input Server Validation

- **CHECK_CRYPTOGRAPHY_AND_VALIDATE_NAME**
CHECK_CRYPTOGRAPHY_ONLY

NetworkInputSettings

Network source to transcode. Must be accessible to the Elemental Live node that is running the live event through a network connection.

hlsInputSettings

Specifies HLS input settings when the uri is for a HLS manifest.

  Type: HlsInputSettings (p. 504)
  Required: False

serverValidation

Check HTTPS server certificates. When set to checkCryptographyOnly, cryptography in the certificate will be checked, but not the server's name. Certain subdomains (notably S3 buckets that use dots in the bucket name) do not strictly match the corresponding certificate's wildcard pattern and would otherwise cause the event to error. This setting is ignored for protocols that do not use https.

  Type: NetworkInputServerValidation (p. 532)
  Required: False

Output

Output settings. There can be multiple outputs within a group.

videoDescriptionName

The name of the VideoDescription used as the source for this output.

  Type: string
  Required: False

outputName

The name used to identify an output.

  Type: string
  Required: False
  MinLength: 1
  MaxLength: 255

captionDescriptionNames

The names of the CaptionDescriptions used as caption sources for this output.

  Type: Array of type string
  Required: False

outputSettings

Output type-specific settings.
Type: OutputSettings (p. 536)
Required: True

**audioDescriptionNames**
The names of the AudioDescriptions used as audio sources for this output.

Type: Array of type string
Required: False

**OutputDestination**

**mediaPackageSettings**
Destination settings for a MediaPackage output; one destination for both encoders.

Type: Array of type MediaPackageOutputDestinationSettings (p. 528)
Required: False

**settings**
Destination settings for a standard output; one destination for each redundant encoder.

Type: Array of type OutputDestinationSettings (p. 534)
Required: False

**id**
User-specified id. This is used in an output group or an output.

Type: string
Required: False

**OutputDestinationSettings**

**passwordParam**
key used to extract the password from EC2 Parameter store

Type: string
Required: False

**streamName**
Stream name for RTMP destinations (URLs of type rtmp://)

Type: string
Required: False

**url**
A URL specifying a destination

Type: string
Properties

Required: False

**username**

Username for destination

Type: string
Required: False

**OutputGroup**

Output groups for this Live Event. Output groups contain information about where streams should be distributed.

**outputs**

Type: Array of type Output (p. 533)
Required: True

**outputGroupSettings**

Settings associated with the output group.

Type: OutputGroupSettings (p. 535)
Required: True

**name**

Custom output group name optionally defined by the user. Only letters, numbers, and the underscore character allowed; only 32 characters allowed.

Type: string
Required: False
MaxLength: 32

**OutputGroupSettings**

Output Group Settings

**archiveGroupSettings**

Type: ArchiveGroupSettings (p. 449)
Required: False

**mediaPackageGroupSettings**

Type: MediaPackageGroupSettings (p. 528)
Required: False

**rtmpGroupSettings**

Type: RtmpGroupSettings (p. 538)
Required: False
udpGroupSettings
Type: UdpGroupSettings (p. 546)
Required: False

msSmoothGroupSettings
Type: MsSmoothGroupSettings (p. 529)
Required: False

hlsGroupSettings
Type: HlsGroupSettings (p. 498)
Required: False

frameCaptureGroupSettings
Type: FrameCaptureGroupSettings (p. 483)
Required: False

OutputLocationRef
Reference to an OutputDestination ID defined in the channel

destinationRefId
Type: string
Required: False

OutputSettings
Output Settings

rtmpOutputSettings
Type: RtmpOutputSettings (p. 539)
Required: False

archiveOutputSettings
Type: ArchiveOutputSettings (p. 449)
Required: False

frameCaptureOutputSettings
Type: FrameCaptureOutputSettings (p. 483)
Required: False

msSmoothOutputSettings
Type: MsSmoothOutputSettings (p. 532)
Required: False
mediaPackageOutputSettings
Type: MediaPackageOutputSettings (p. 529)
Required: False

udpOutputSettings
Type: UdpOutputSettings (p. 547)
Required: False

hlsOutputSettings
Type: HlsOutputSettings (p. 506)
Required: False

PassThroughSettings
Pass Through Settings

RemixSettings
Remix Settings

channelMappings
Mapping of input channels to output channels, with appropriate gain adjustments.
Type: Array of type AudioChannelMapping (p. 450)
Required: True

channelsOut
Number of output channels to be produced. Valid values: 1, 2, 4, 6, 8
Type: integer
Required: False
Minimum: 1
Maximum: 8

channelsIn
Number of input channels to be used.
Type: integer
Required: False
Minimum: 1
Maximum: 16

ResourceConflict
message
Type: string
Required: False
ResourceNotFound

message

  Type: string
  Required: False

RtmpCacheFullBehavior

Rtmp Cache Full Behavior

  DISCONNECT_IMMEDIATELY
  WAIT_FOR_SERVER

RtmpCaptionData

Rtmp Caption Data

  ALL
  FIELD1_608
  FIELD1_AND_FIELD2_608

RtmpCaptionInfoDestinationSettings

Rtmp Caption Info Destination Settings

RtmpGroupSettings

Rtmp Group Settings

inputLossAction

Controls the behavior of this RTMP group if input becomes unavailable. - emitOutput: Emit a slate until input returns. - pauseOutput: Stop transmitting data until input returns. This does not close the underlying RTMP connection.

  Type: InputLossActionForRtmpOut (p. 511)
  Required: False

captionData

Controls the types of data that passes to onCaptionInfo outputs. If set to 'all' then 608 and 708 carried DTVCC data will be passed. If set to 'field1AndField2608' then DTVCC data will be stripped out, but 608 data from both fields will be passed. If set to 'field1608' then only the data carried in 608 from field 1 video will be passed.

  Type: RtmpCaptionData (p. 538)
  Required: False

authenticationScheme

Authentication scheme to use when connecting with CDN
**Properties**

**Type**
AuthenticationScheme (p. 456)

*Required*: False

**cacheLength**

Cache length, in seconds, is used to calculate buffer size.

*Type*: integer

*Required*: False

*Minimum*: 30

**restartDelay**

If a streaming output fails, number of seconds to wait until a restart is initiated. A value of 0 means never restart.

*Type*: integer

*Required*: False

*Minimum*: 0

**cacheFullBehavior**

Controls behavior when content cache fills up. If remote origin server stalls the RTMP connection and does not accept content fast enough the 'Media Cache' will fill up. When the cache reaches the duration specified by cacheLength the cache will stop accepting new content. If set to disconnectImmediately, the RTMP output will force a disconnect. Clear the media cache, and reconnect after restartDelay seconds. If set to waitForServer, the RTMP output will wait up to 5 minutes to allow the origin server to begin accepting data again.

*Type*: RtmpCacheFullBehavior (p. 538)

*Required*: False

**RtmpOutputCertificateMode**

Rtmp Output Certificate Mode

SELF_SIGNED
VERIFY_AUTHENTICITY

**RtmpOutputSettings**

Rtmp Output Settings

**CertificateMode**

If set to verifyAuthenticity, verify the tls certificate chain to a trusted Certificate Authority (CA). This will cause rtmps outputs with self-signed certificates to fail.

*Type*: RtmpOutputCertificateMode (p. 539)

*Required*: False

**numRetries**

Number of retry attempts.
**Properties**

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

**destination**

The RTMP endpoint excluding the stream name (e.g. rtmp://host/appname). For connection to Akamai, a username and password must be supplied. URI fields accept format identifiers.

- **Type**: OutputLocationRef (p. 536)  
  **Required**: True

**connectionRetryInterval**

Number of seconds to wait before retrying a connection to the Flash Media server if the connection is lost.

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

**Scte20Convert608To708**

Scte20 Convert608 To708

- **DISABLED**  
  - **UPCONVERT**

**Scte20PlusEmbeddedDestinationSettings**

Scte20 Plus Embedded Destination Settings

**Scte20SourceSettings**

Scte20 Source Settings

- **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**

  If 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.

  - **Type**: Scte20Convert608To708 (p. 540)  
    - **Required**: False
**Scte27DestinationSettings**

Scte27 Destination Settings

**Scte27SourceSettings**

Scte27 Source Settings

**pid**

The `pid` field is used in conjunction with the caption selector languageCode field as follows:
- Specify PID and Language: Extracts captions from that PID; the language is "informational".
- Specify PID and omit Language: Extracts the specified PID.
- Omit PID and specify Language: Extracts the specified language, whichever PID that happens to be.
- Omit PID and omit Language: Valid only if source is DVB-Sub that is being passed through; all languages will be passed through.

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

**Scte35AposNoRegionalBlackoutBehavior**

Scte35 Apos No Regional Blackout Behavior

  - FOLLOW
  - IGNORE

**Scte35AposWebDeliveryAllowedBehavior**

Scte35 Apos Web Delivery Allowed Behavior

  - FOLLOW
  - IGNORE

**Scte35SpliceInsert**

Scte35 Splice Insert

**adAvailOffset**

When specified, this offset (in milliseconds) is added to the input Ad Avail PTS time. This only applies to embedded SCTE 104/35 messages and does not apply to OOB messages.

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

**webDeliveryAllowedFlag**

When set to ignore, Segment Descriptors with `webDeliveryAllowedFlag` set to 0 will no longer trigger blackouts or Ad Avail slates.
Properties

**noRegionalBlackoutFlag**

When set to ignore, Segment Descriptors with noRegionalBlackoutFlag set to 0 will no longer trigger blackouts or Ad Avail slates

- **Type**: `Scte35SpliceInsertNoRegionalBlackoutBehavior` (p. 542)
- **Required**: False

**Scte35SpliceInsertNoRegionalBlackoutBehavior**

Scte35 Splice Insert No Regional Blackout Behavior

- FOLLOW
- IGNORE

**Scte35SpliceInsertWebDeliveryAllowedBehavior**

Scte35 Splice Insert Web Delivery Allowed Behavior

- FOLLOW
- IGNORE

**Scte35TimeSignalApos**

Scte35 Time Signal Apos

**adAvailOffset**

When specified, this offset (in milliseconds) is added to the input Ad Avail PTS time. This only applies to embedded SCTE 104/35 messages and does not apply to OOB messages.

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

**webDeliveryAllowedFlag**

When set to ignore, Segment Descriptors with webDeliveryAllowedFlag set to 0 will no longer trigger blackouts or Ad Avail slates

- **Type**: `Scte35AposWebDeliveryAllowedBehavior` (p. 541)
- **Required**: False

**noRegionalBlackoutFlag**

When set to ignore, Segment Descriptors with noRegionalBlackoutFlag set to 0 will no longer trigger blackouts or Ad Avail slates

- **Type**: `Scte35AposNoRegionalBlackoutBehavior` (p. 541)
**Required:** False

**SmoothGroupAudioOnlyTimecodeControl**

Smooth Group Audio Only Timecode Control

- PASSTHROUGH
- USE_CONFIGURED_CLOCK

**SmoothGroupCertificateMode**

Smooth Group Certificate Mode

- SELF_SIGNED
- VERIFY_AUTHENTICITY

**SmoothGroupEventIdMode**

Smooth Group Event Id Mode

- NO_EVENT_ID
- USE_CONFIGURED
- USE_TIMESTAMP

**SmoothGroupEventStopBehavior**

Smooth Group Event Stop Behavior

- NONE
- SEND_EOS

**SmoothGroupSegmentationMode**

Smooth Group Segmentation Mode

- USE_INPUT_SEGMENTATION
- USE_SEGMENT_DURATION

**SmoothGroupSparseTrackType**

Smooth Group Sparse Track Type

- NONE
- SCTE_35

**SmoothGroupStreamManifestBehavior**

Smooth Group Stream Manifest Behavior

- DO_NOT_SEND
- SEND
**SmoothGroupTimestampOffsetMode**

Smooth Group Timestamp Offset Mode

- USE_CONFIGURED_OFFSET
- USE_EVENT_START_DATE

**SmpteTtDestinationSettings**

Smpte Tt Destination Settings

**StandardHlsSettings**

Standard Hls Settings

**m3u8Settings**

Type: M3u8Settings (p. 525)

Required: True

**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

**StaticKeySettings**

Static Key Settings

**staticKeyValue**

Static key value as a 32 character hexadecimal string.

Type: string

Required: True

MinLength: 32

MaxLength: 32

**keyProviderServer**

The URL of the license server used for protecting content.

Type: InputLocation (p. 510)

Required: False

**Tags**

key-value pairs

Type: string
**TeletextDestinationSettings**

Teletext Destination Settings

**TeletextSourceSettings**

Teletext Source Settings

**pageNumber**

Specifies the teletext page number within the data stream from which to extract captions. Range of 0x100 (256) to 0x8FF (2303). Unused for passthrough. Should be specified as a hexadecimal string with no "0x" prefix.

*Type: string*

*Required: False*

**TimecodeConfig**

Timecode Config

**syncThreshold**

Threshold in frames beyond which output timecode is resynchronized to the input timecode. Discrepancies below this threshold are permitted to avoid unnecessary discontinuities in the output timecode. No timecode sync when this is not specified.

*Type: integer*

*Required: False*

*Minimum: 1*

*Maximum: 1000000*

**source**

Identifies the source for the timecode that will be associated with the events outputs. - Embedded (embedded): Initialize the output timecode with timecode from the the source. If no embedded timecode is detected in the source, the system falls back to using "Start at 0" (zerobased). - System Clock (systemclock): Use the UTC time. - Start at 0 (zerobased): The time of the first frame of the event will be 00:00:00:00.

*Type: TimecodeConfigSource (p. 545)*

*Required: True*

**TimecodeConfigSource**

Timecode Config Source

- EMBEDDED
- SYSTEMCLOCK
- ZEROBASED

**TtmlDestinationSettings**

Ttml Destination Settings
**styleControl**

When set to passthrough, passes through style and position information from a TTML-like input source (TTML, SMPTE-TT, CFF-TT) to the CFF-TT output or TTML output.

*Type:* TtmlDestinationStyleControl (p. 546)

*Required:* False

---

**TtmlDestinationStyleControl**

Ttml Destination Style Control

- PASSTHROUGH
- USE_CONFIGURED

---

**UdpContainerSettings**

Udp Container Settings

---

**m2tsSettings**

*Type:* M2tsSettings (p. 518)

*Required:* False

---

**UdpGroupSettings**

Udp Group Settings

---

**inputLossAction**

Specifies behavior of last resort when input video is lost, and no more backup inputs are available. When dropTs is selected the entire transport stream will stop being emitted. When dropProgram is selected the program can be dropped from the transport stream (and replaced with null packets to meet the TS bitrate requirement). Or, when emitProgram is chosen the transport stream will continue to be produced normally with repeat frames, black frames, or slate frames substituted for the absent input video.

*Type:* InputLossActionForUdpOut (p. 511)

*Required:* False

---

**timedMetadataId3Frame**

Indicates ID3 frame that has the timecode.

*Type:* UdpTimedMetadataId3Frame (p. 547)

*Required:* False

---

**timedMetadataId3Period**

Timed Metadata interval in seconds.

*Type:* integer

*Required:* False
Minimum: 0

**UdpOutputSettings**

Udp Output Settings

**destination**

Destination address and port number for RTP or UDP packets. Can be unicast or multicast RTP or UDP (eg. rtp://239.10.10.10:5001 or udp://10.100.100.100:5002).

Type: OutputLocationRef (p. 536)
Required: True

**bufferMsec**

UDP output buffering in milliseconds. Larger values increase latency through the transcoder but simultaneously assist the transcoder in maintaining a constant, low-jitter UDP/RTP output while accommodating clock recovery, input switching, input disruptions, picture reordering, etc.

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

**containerSettings**

Type: UdpContainerSettings (p. 546)
Required: True

**fecOutputSettings**

Settings for enabling and adjusting Forward Error Correction on UDP outputs.

Type: FecOutputSettings (p. 482)
Required: False

**UdpTimedMetadataId3Frame**

Udp Timed Metadata Id3 Frame

NONE
PRIV
TDRL

**VideoCodecSettings**

Video Codec Settings

**h264Settings**

Type: H264Settings (p. 488)
Properties

**Required:** False

**frameCaptureSettings**

Type: FrameCaptureSettings (p. 484)

**Required:** False

**VideoDescription**

Video settings for this stream.

**respondToAfd**

Indicates how to respond to the AFD values in the input stream. RESPOND causes input video to be clipped, depending on the AFD value, input display aspect ratio, and output display aspect ratio, and (except for FRAMECAPTURE codec) includes the values in the output. PASSTHROUGH (does not apply to FRAMECAPTURE codec) ignores the AFD values and includes the values in the output, so input video is not clipped. NONE ignores the AFD values and does not include the values through to the output, so input video is not clipped.

Type: VideoDescriptionRespondToAfd (p. 549)

**Required:** False

**scalingBehavior**

STRETCHTOOUTPUT configures the output position to stretch the video to the specified output resolution (height and width). This option will override any position value. DEFAULT may insert black boxes (pillar boxes or letter boxes) around the video to provide the specified output resolution.

Type: VideoDescriptionScalingBehavior (p. 549)

**Required:** False

**name**

The name of this VideoDescription. Outputs will use this name to uniquely identify this Description. Description names should be unique within this Live Event.

Type: string

**Required:** True

**width**

Output video width, in pixels. Must be an even number. For most codecs, you can leave this field and height blank in order to use the height and width (resolution) from the source. Note, however, that leaving blank is not recommended. For the Frame Capture codec, height and width are required.

Type: integer

**Required:** False

**sharpness**

Changes the strength of the anti-alias filter used for scaling. 0 is the softest setting, 100 is the sharpest. A setting of 50 is recommended for most content.
Properties

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

**codecSettings**

Video codec settings.

- **Type**: VideoCodecSettings (p. 547)
- **Required**: False

**height**

Output video height, in pixels. Must be an even number. For most codecs, you can leave this field and width blank in order to use the height and width (resolution) from the source. Note, however, that leaving blank is not recommended. For the Frame Capture codec, height and width are required.

- **Type**: integer
- **Required**: False

**VideoDescriptionRespondToAfd**

Video Description Respond To Afd

- NONE
- PASSTHROUGH
- RESPOND

**VideoDescriptionScalingBehavior**

Video Description Scaling Behavior

- DEFAULT
- STRETCH_TO_OUTPUT

**VideoSelector**

Specifies a particular video stream within an input source. An input may have only a single video selector.

**colorSpace**

Specifies the colorspace of an input. This setting works in tandem with colorSpaceConversion to determine if any conversion will be performed.

- **Type**: VideoSelectorColorSpace (p. 550)
- **Required**: False

**selectorSettings**

The video selector settings.

- **Type**: VideoSelectorSettings (p. 551)
Required: False

colorSpaceUsage
Applies only if colorSpace is a value other than follow. This field controls how the value in the colorSpace field will be used. fallback means that when the input does include color space data, that data will be used, but when the input has no color space data, the value in colorSpace will be used. Choose fallback if your input is sometimes missing color space data, but when it does have color space data, that data is correct. force means to always use the value in colorSpace. Choose force if your input usually has no color space data or might have unreliable color space data.

Type: VideoSelectorColorSpaceUsage (p. 550)
Required: False

VideoSelectorColorSpace
Video Selector Color Space
- FOLLOW
- REC_601
- REC_709

VideoSelectorColorSpaceUsage
Video Selector Color Space Usage
- FALLBACK
- FORCE

VideoSelectorPid
Video Selector Pid
pid
Selects a specific PID from within a video source.

Type: integer
Required: False
Minimum: 0
Maximum: 8191

VideoSelectorProgramId
Video Selector Program Id
programId
Selects a specific program from within a multi-program transport stream. If the program doesn't exist, the first program within the transport stream will be selected by default.

Type: integer
Required: False
Minimum: 0
Channels channelId Stop

URI

/prod/channels/{channelId}/stop

HTTP Methods

POST

Operation ID: StopChannel

Stops a running channel

Path Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>channelId</td>
<td>String</td>
<td>True</td>
<td>channel ID</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>Channel (p. 552)</td>
<td>Successfully initiated stop of the channel.</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 563)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 563)</td>
<td>You do not have permission to list channels.</td>
</tr>
</tbody>
</table>
Status Code | Response Model | Description
--- | --- | ---
404 | ResourceNotFound (p. 563) | The channel you're requesting to describe does not exist.
409 | ResourceConflict (p. 564) | The channel is unable to create due to an issue with channel resources.
429 | LimitExceeded (p. 564) | Request limit exceeded on list channel calls to channel service.
500 | InternalServiceError (p. 564) | Unexpected internal service error.
502 | BadGatewayException (p. 564) | Bad Gateway Error
504 | GatewayTimeoutException (p. 565) | Gateway Timeout Error

Schemas

Response Bodies

Example Channel

```json
{
    "inputAttachments": [
        {
            "inputId": "string",
            "inputAttachmentName": "string",
            "inputSettings": {
                "sourceEndBehavior": enum,
                "deblockFilter": enum,
                "audioSelectors": [
                    {
                        "name": "string",
                        "selectorSettings": {
                            "audioLanguageSelection": {
                                "languageSelectionPolicy": enum,
                                "languageCode": "string"
                            },
                            "audioPidSelection": {
                                "pid": integer
                            }
                        }
                    }
                ],
                "networkInputSettings": {
                    "hlsInputSettings": {
                        "retries": integer,
                        "bandwidth": integer,
                        "retryInterval": integer,
                        "bufferSegments": integer
                    },
                    "serverValidation": enum
                },
                "inputFilter": enum,
                "videoSelector": {
                    "colorSpace": enum,
                    "selectorSettings": {
```
"videoSelectorPid": {
  "pid": integer
},
"videoSelectorProgramId": {
  "programId": integer
},
"colorSpaceUsage": enum,
"filterStrength": integer,
"denoiseFilter": enum,
"captionSelectors": [
  {
    "name": "string",
    "languageCode": "string",
    "selectorSettings": {
      "embeddedSourceSettings": {
        "scte20Detection": enum,
        "source608ChannelNumber": integer,
        "convert608To708": enum,
        "source608TrackNumber": integer
      },
      "scte20SourceSettings": {
        "source608ChannelNumber": integer,
        "convert608To708": enum
      },
      "dvbSubSourceSettings": {
        "pid": integer
      },
      "teletextSourceSettings": {
        "pageNumber": "string"
      },
      "aribSourceSettings": {
      },
      "scte27SourceSettings": {
        "pid": integer
      }
    }
  }
],
"destinations": [
  {
    "mediaPackageSettings": [
      {
        "channelId": "string"
      }
    ],
    "settings": [
      {
        "passwordParam": "string",
        "streamName": "string",
        "url": "string",
        "username": "string"
      }
    ],
    "id": "string"
  }
],
"encoderSettings": {
  "timecodeConfig": {
    "syncThreshold": integer,
    "source": enum
  }
}
"outputGroups": [
  {
    "outputs": [
      {
        "videoDescriptionName": "string",
        "outputName": "string",
        "captionDescriptionNames": [
          "string"
        ],
        "outputSettings": {
          "rtmpOutputSettings": {
            "certificateMode": enum,
            "numRetries": integer,
            "destination": {
              "destinationRefId": "string"
            },
            "connectionRetryInterval": integer
          },
          "archiveOutputSettings": {
            "extension": "string",
            "containerSettings": {
              "m2tsSettings": {
                "audioStreamType": enum,
                "ecmPid": "string",
                "dvbTeletextPid": "string",
                "aribCaptionsPidControl": enum,
                "bitrate": integer,
                "rateMode": enum,
                "segmentationTime": number,
                "audioPids": "string",
                "audioFramesPerPes": integer,
                "fragmentTime": number,
                "ebpLookaheadMs": integer,
                "ebpAudioInterval": enum,
                "scte35Pid": "string",
                "programNum": integer,
                "pmtInterval": integer,
                "pcrPeriod": integer,
                "segmentationStyle": enum,
                "ebif": enum,
                "audioBufferModel": enum,
                "dvbNitSettings": {
                  "networkName": "string",
                  "networkId": integer,
                  "repInterval": integer
                },
                "absentInputAudioBehavior": enum,
                "timedMetadataBehavior": enum,
                "pmtPid": "string",
                "etvSignalPid": "string",
                "bufferModel": enum,
                "scte35Control": enum,
                "ebpPlacement": enum,
                "arib": enum,
                "nullPacketBitrate": number,
                "dvbSdtSettings": {
                  "serviceName": "string",
                  "serviceProviderName": "string",
                  "repInterval": integer,
                  "outputSdt": enum
                },
                "pcrPid": "string",
                "transportStreamId": integer,
                "pcrControl": enum,
                "videoPid": "string"
              }
            }
          }
        }
      }
    }
  }
]
"esRateInPes": enum,
"segmentationMarkers": enum,
"dvbTdtSettings": {
    "repInterval": integer
},
"klv": enum,
"ccDescriptor": enum,
"patInterval": integer,
"etvPlatformPid": "string",
"dvbSubPids": "string",
"aribCaptionsPid": "string",
"scte27Pids": "string",
"klvDataPids": "string"
},
"nameModifier": "string"
},
"frameCaptureOutputSettings": {
"nameModifier": "string"
},
"msSmoothOutputSettings": {
"nameModifier": "string"
},
"mediaPackageOutputSettings": {
},
"udpOutputSettings": {
"destination": {
    "destinationRefId": "string"
},
"bufferMsec": integer,
"containerSettings": {
    "m2tsSettings": {
        "audioStreamType": enum,
        "ecmPid": "string",
        "dvbTeletextPid": "string",
        "aribCaptionsPidControl": enum,
        "bitrate": integer,
        "rateMode": enum,
        "segmentationTime": number,
        "audioPids": "string",
        "audioFramesPerPes": integer,
        "fragmentTime": number,
        "ebpLookaheadMbs": integer,
        "ebpAudioInterval": enum,
        "scte35Pid": "string",
        "programNum": integer,
        "pmtpInterval": integer,
        "pcrPeriod": integer,
        "segmentationStyle": enum,
        "ebif": enum,
        "audioBufferModel": enum,
        "dvbNitSettings": {
            "networkName": "string",
            "networkId": integer,
            "repInterval": integer
        },
        "absentInputAudioBehavior": enum,
        "timedMetadataBehavior": enum,
        "timedMetadataPid": "string",
        "pmtp": "string",
        "etvSignalPid": "string",
        "bufferModel": enum,
        "scte35Control": enum,
        "ebpPlacement": enum,
        "arib": enum,
        "nullPacketBitrate": number,
"dvbSdtSettings": {  
  "serviceName": "string",  
  "serviceProviderName": "string",  
  "repInterval": integer,  
  "outputSdt": enum  },  
"pcrPid": "string",  
"transportStreamId": integer,  
"pcrControl": enum,  
"videoPid": "string",  
"esRateInPes": enum,  
"segmentationMarkers": enum,  
"dvbTdtSettings": {  
  "repInterval": integer  },  
"klv": enum,  
"ccDescriptor": enum,  
"patInterval": integer,  
"etvPlatformPid": "string",  
"dvbSubPids": "string",  
"aribCaptionsPid": "string",  
"scte27Pids": "string",  
"klvDataPids": "string"  }  }
"fecOutputSettings": {  
  "rowLength": integer,  
  "columnDepth": integer,  
  "includeFec": enum  }  }
"hlsOutputSettings": {  
  "segmentModifier": "string",  
  "hlsSettings": {  
  "standardHlsSettings": {  
  "m3u8Settings": {  
  "pmtPid": "string",  
  "ecmPid": "string",  
  "scte35Behavior": enum,  
  "pcrPid": "string",  
  "audioPids": "string",  
  "audioFramesPerPes": integer,  
  "scte35Pid": "string",  
  "transportStreamId": integer,  
  "pcrControl": enum,  
  "videoPid": "string",  
  "pcrPeriod": integer,  
  "pmtInterval": integer,  
  "programNum": integer,  
  "patInterval": integer,  
  "timedMetadataPid": "string",  
  "timedMetadataBehavior": enum  }  },  
  "audioRenditionSets": "string"  }  },  
  "audioOnlyHlsSettings": {  
  "audioTrackType": enum,  
  "audioGroupId": "string",  
  "audioOnlyImage": {  
  "passwordParam": "string",  
  "uri": "string",  
  "username": "string"  }  }  }  }
"nameModifier": "string"
"audioDescriptionNames": [ "string"
],
"outputGroupSettings": {
  "archiveGroupSettings": {
    "destination": {
      "destinationRefId": "string"
    },
    "rolloverInterval": integer
  },
  "mediaPackageGroupSettings": {
    "destination": {
      "destinationRefId": "string"
    }
  },
  "rtmpGroupSettings": {
    "inputLossAction": enum,
    "captionData": enum,
    "authenticationScheme": enum,
    "cacheLength": integer,
    "restartDelay": integer,
    "cacheFullBehavior": enum
  },
  "udpGroupSettings": {
    "inputLossAction": enum,
    "timedMetadataId3Frame": enum,
    "timedMetadataId3Period": integer
  },
  "msSmoothGroupSettings": {
    "fragmentLength": integer,
    "eventId": "string",
    "timestampOffset": "string",
    "segmentationMode": enum,
    "numRetries": integer,
    "eventStopBehavior": enum,
    "acquisitionPointId": "string",
    "sparseTrackType": enum,
    "timestampOffsetMode": enum,
    "destination": {
      "destinationRefId": "string"
    },
    "audioOnlyTimecodeControl": enum,
    "connectionRetryInterval": integer,
    "filecacheDuration": integer,
    "certificateMode": enum,
    "inputLossAction": enum,
    "sendDelayMs": integer,
    "eventIdMode": enum,
    "restartDelay": integer,
    "streamManifestBehavior": enum
  },
  "hlsGroupSettings": {
    "segmentsPerSubdirectory": integer,
    "l1vInManifest": enum,
    "outputSelection": enum,
    "destination": {
      "destinationRefId": "string"
    },
    "encryptionType": enum,
    "indexNSegments": integer,
    "constantIV": "string",
    "timedMetadataId3Frame": enum,
"baseUrlManifest": "string",
"captionLanguageSetting": enum,
"minSegmentLength": integer,
"mode": enum,
"keyProviderSettings": {
  "staticKeySettings": {
    "staticKeyValue": "string",
    "keyProviderServer": {
      "passwordParam": "string",
      "uri": "string",
      "username": "string"
    }
  }
},
"manifestCompression": enum,
"ivSource": enum,
"tsFileMode": enum,
"manifestDurationFormat": enum,
"keyFormatVersions": "string",
"streamInfResolution": enum,
"timestampDeltaMilliseconds": integer,
"baseUrlContent": "string",
"manifestDurationFormat": enum,
"segmentationMode": enum,
"captionLanguageMappings": [
  {
    "languageDescription": "string",
    "captionChannel": integer,
    "languageCode": "string"
  }
],
"clientCache": enum,
"codecSpecification": enum,
"keepSegments": integer,
"redundantManifest": enum,
"timedMetadataId3Period": integer,
"programDateTime": enum,
"directoryStructure": enum,
"keyFormat": "string",
"inputLossAction": enum,
"adMarkers": [enum
],
"programDateTimePeriod": integer,
"segmentLength": integer,
"hlsCdnSettings": {
  "hlsAkamaiSettings": {
    "salt": "string",
    "httpTransferMode": enum,
    "numRetries": integer,
    "restartDelay": integer,
    "connectionRetryInterval": integer,
    "filecacheDuration": integer,
    "token": "string"
  },
  "hlsWebdavSettings": {
    "httpTransferMode": enum,
    "numRetries": integer,
    "restartDelay": integer,
    "connectionRetryInterval": integer,
    "filecacheDuration": integer
  },
  "hlsBasicPutSettings": {
    "numRetries": integer,
    "restartDelay": integer,
    "connectionRetryInterval": integer,
    "filecacheDuration": integer
  }
}
"hlsMediaStoreSettings": {
"mediaStoreStorageClass": enum,
"numRetries": integer,
"restartDelay": integer,
"connectionRetryInterval": integer,
"filecacheDuration": integer
},
"iFrameOnlyPlaylists": enum
},
"frameCaptureGroupSettings": {
"destination": {
"destinationRefId": "string"
}
},
"name": "string"
],
"audioDescriptions": [
{
"audioTypeControl": enum,
"languageCodeControl": enum,
"remixSettings": {
"channelMappings": [
{
"outputChannel": integer,
"inputChannelLevels": [
{
"inputChannel": integer,
"gain": integer
}
]
}
].
"channelsOut": integer,
"channelsIn": integer
},
"audioType": enum,
"name": "string",
"codecSettings": {
"aacSettings": {
"vbrQuality": enum,
"codingMode": enum,
"profile": enum,
"inputType": enum,
"bitrate": number,
"rawFormat": enum,
"rateControlMode": enum,
"sampleRate": number,
"spec": enum
},
"ac3Settings": {
"drcProfile": enum,
"dialnorm": integer,
"codingMode": enum,
"metadataControl": enum,
"bitrate": number,
"lfeFilter": enum,
"bitstreamMode": enum
},
"eac3Settings": {
"dialnorm": integer,
"passthroughControl": enum,
"drcLine": enum,
"spec": enum
}
"metadataControl": enum,
"bitrate": number,
"ltRtSurroundMixLevel": number,
"surroundExMode": enum,
"lfeControl": enum,
"codingMode": enum,
"surroundMode": enum,
"attenuationControl": enum,
"lfeFilter": enum,
"dcFilter": enum,
"ltRtCenterMixLevel": number,
"phaseControl": enum,
"bitstreamMode": enum,
"stereoDownmix": enum,
"loRoSurroundMixLevel": number,
"drcRf": enum,
"loRoCenterMixLevel": number
},
"passThroughSettings": {
},
"mp2Settings": {
 "codingMode": enum,
 "bitrate": number,
 "sampleRate": number
}
},
"languageCode": "string",
"streamName": "string",
"audioNormalizationSettings": {
 "targetLkfs": number,
 "algorithmControl": enum,
 "algorithm": enum
},
"audioSelectorName": "string"
],
"availConfiguration": {
 "availSettings": {
 "scte35TimeSignalApos": {
 "adAvailOffset": integer,
 "webDeliveryAllowedFlag": enum,
 "noRegionalBlackoutFlag": enum
 },
 "scte35SpliceInsert": {
 "adAvailOffset": integer,
 "webDeliveryAllowedFlag": enum,
 "noRegionalBlackoutFlag": enum
 }
 },
"captionDescriptions": [
 {
 "captionSelectorName": "string",
 "languageDescription": "string",
 "name": "string",
 "languageCode": "string",
 "destinationSettings": {
 "burnInDestinationSettings": {
 "xPosition": integer,
 "backgroundColor": enum,
 "yPosition": integer,
 "teletextGridControl": enum,
 "backgroundOpacity": integer,
 "fontOpacity": integer,
 "fontResolution": integer,
 "shadowOpacity": integer,
"shadowYOffset": integer,
"outlineSize": integer,
"outlineColor": enum,
"fontSize": "string",
"alignment": enum,
"shadowXOffset": integer,
"shadowColor": enum,
"fontColor": enum,
"font": {
  "passwordParam": "string",
  "uri": "string",
  "username": "string"
}
},
"scte27DestinationSettings": {
},
"teletextDestinationSettings": {
},
"ttmlDestinationSettings": {
  "styleControl": enum
},
"smpteTtDestinationSettings": {
},
"webvttDestinationSettings": {
},
"embeddedPlusScte20DestinationSettings": {
},
"dvbSubDestinationSettings": {
  "xPosition": integer,
  "backgroundColor": enum,
  "yPosition": integer,
  "teletextGridControl": enum,
  "backgroundColor": integer,
  "fontOpacity": integer,
  "fontResolution": integer,
  "shadowOpacity": integer,
  "shadowYOffset": integer,
  "outlineSize": integer,
  "outlineColor": enum,
  "fontSize": "string",
  "alignment": enum,
  "shadowXOffset": integer,
  "shadowColor": enum,
  "fontColor": enum,
  "font": {
    "passwordParam": "string",
    "uri": "string",
    "username": "string"
  }
},
"embeddedDestinationSettings": {
},
"rtmpCaptionInfoDestinationSettings": {
},
"aribDestinationSettings": {
},
"scte20PlusEmbeddedDestinationSettings": {
}
},
"globalConfiguration": {
  "inputLossBehavior": {
    "inputLossImageType": enum,
    "inputLossImageColor": "string",
    "inputLossImageSlate": {
"passwordParam": "string",
"uri": "string",
"username": "string"
],
"repeatFrameMsec": integer,
"blackFrameMsec": integer
],
"supportLowFramerateInputs": enum,
"outputLockingMode": enum,
"initialAudioGain": integer,
"inputEndAction": enum,
"outputTimingSource": enum
],
"videoDescriptions": [ ]
{
  "respondToAfd": enum,
  "scalingBehavior": enum,
  "name": "string",
  "width": integer,
  "sharpness": integer,
  "codecSettings": {
    "h264Settings": {
      "minIInterval": integer,
      "slices": integer,
      "parNumerator": integer,
      "gopSizeUnits": enum,
      "subgopLength": enum,
      "maxBitrate": integer,
      "bitrate": integer,
      "bufFillPct": integer,
      "temporalAq": enum,
      "afdSignaling": enum,
      "timecodeInsertion": enum,
      "bufSize": integer,
      "softness": integer,
      "framerateControl": enum,
      "gbrQualityLevel": integer,
      "fixedAfd": enum,
      "level": enum,
      "lookAheadRateControl": enum,
      "profile": enum,
      "framerateNumerator": integer,
      "gopClosedCadence": integer,
      "entropyEncoding": enum,
      "framerateDenominator": integer,
      "spatialAq": enum,
      "adaptiveQuantization": enum,
      "colorMetadata": enum,
      "gopSize": number,
      "numRefFrames": integer,
      "gopBReference": enum,
      "parControl": enum,
      "parDenominator": integer,
      "syntax": enum,
      "sceneChangeDetect": enum,
      "scanType": enum,
      "flickerAq": enum,
      "gopNumBFrames": integer,
      "rateControlMode": enum
    },
    "frameCaptureSettings": {
      "captureInterval": integer
    }
  }
],
"height": integer
}
Schemas

```
"availability": {  
  "state": enum,  
  "availabilityImage": {  
    "passwordParam": "string",  
    "uri": "string",  
    "username": "string"  
  },  
  "blackoutSlate": {  
    "networkEndBlackoutImage": {  
      "passwordParam": "string",  
      "uri": "string",  
      "username": "string"  
    },  
    "networkEndBlackout": enum,  
    "networkId": "string",  
    "state": enum,  
    "blackoutSlateImage": {  
      "passwordParam": "string",  
      "uri": "string",  
      "username": "string"  
    }  
  },  
  "egressEndpoints": [  
    {  
      "sourceIp": "string"  
    }  
  ],  
  "inputSpecification": {  
    "codec": enum,  
    "resolution": enum,  
    "maximumBitrate": enum  
  },  
  "channelClass": enum,  
  "tags": {  
  },  
  "logLevel": enum,  
  "roleArn": "string",  
  "name": "string",  
  "id": "string",  
  "state": enum,  
  "pipelinesRunningCount": integer,  
  "arn": "string"  
}
```

**Example InvalidRequest**

```
{  
  "message": "string"  
}
```

**Example AccessDenied**

```
{  
  "message": "string"  
}
```

**Example ResourceNotFoundException**

```
{  
}
```
"message": "string"
}

Example ResourceConflict

{
  "message": "string"
}

Example LimitExceeded

{
  "message": "string"
}

Example InternalServiceError

{
  "message": "string"
}

Example BadGatewayException

{
  "message": "string"
}

Example GatewayTimeoutException

{
  "message": "string"
}

Properties

AacCodingMode

Aac Coding Mode

  AD_RECEIVER_MIX
  CODING_MODE_1_0
  CODING_MODE_1_1
  CODING_MODE_2_0
  CODING_MODE_5_1

AacInputType

Aac Input Type

  BROADCASTER_MIXED_AD
  NORMAL
**AacProfile**

Aac Profile

- HEV1
- HEV2
- LC

**AacRateControlMode**

Aac Rate Control Mode

- CBR
- VBR

**AacRawFormat**

Aac Raw Format

- LATM_LOAS
- NONE

**AacSettings**

Aac Settings

**vbrQuality**

VBR Quality Level - Only used if rateControlMode is VBR.

- **Type**: AacVbrQuality (p. 566)
- **Required**: False

**codingMode**

Mono, Stereo, or 5.1 channel layout. Valid values depend on rate control mode and profile. The adReceiverMix 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.

- **Type**: AacCodingMode (p. 564)
- **Required**: False

**profile**

AAC Profile.

- **Type**: AacProfile (p. 565)
- **Required**: False

**inputType**

Set to "broadcasterMixedAd" when input contains pre-mixed main audio + AD (narration) as a stereo pair. The Audio Type field (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. The values in audioTypeControl and audioType (in AudioDescription) are ignored when set to broadcasterMixedAd. Leave set to "normal" when input does not contain pre-mixed audio + AD.

**Type**: AacInputType (p. 564)
**Required**: False

### bitrate
Average bitrate in bits/second. Valid values depend on rate control mode and profile.

**Type**: number
**Required**: False

### rawFormat
Sets LATM / LOAS AAC output for raw containers.

**Type**: AacRawFormat (p. 565)
**Required**: False

### rateControlMode
Rate Control Mode.

**Type**: AacRateControlMode (p. 565)
**Required**: False

### sampleRate
Sample rate in Hz. Valid values depend on rate control mode and profile.

**Type**: number
**Required**: False

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

**Type**: AacSpec (p. 566)
**Required**: False

#### AacSpec

**Aac Spec**

- MPEG2
- MPEG4

#### AacVbrQuality

**Aac Vbr Quality**
HIGH
LOW
MEDIUM_HIGH
MEDIUM_LOW

**Ac3BitstreamMode**

Ac3 Bitstream Mode

COMMENTARY
COMPLETE_MAIN
DIALOGUE
EMERGENCY
HEARING_IMPAIRED
MUSIC_AND_EFFECTS
VISUALLY_IMPAIRED
VOICE_OVER

**Ac3CodingMode**

Ac3 Coding Mode

CODING_MODE_1_0
CODING_MODE_1_1
CODING_MODE_2_0
CODING_MODE_3_2_LFE

**Ac3DrcProfile**

Ac3 Drc Profile

FILM_STANDARD
NONE

**Ac3LfeFilter**

Ac3 Lfe Filter

DISABLED
ENABLED

**Ac3MetadataControl**

Ac3 Metadata Control

FOLLOW_INPUT
USE_CONFIGURED

**Ac3Settings**

Ac3 Settings
**drcProfile**

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

- **Type:** Ac3DrcProfile (p. 567)
- **Required:** False

**dialnorm**

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

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

**codingMode**

Dolby Digital coding mode. Determines number of channels.

- **Type:** Ac3CodingMode (p. 567)
- **Required:** False

**metadataControl**

When set to "followInput", 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.

- **Type:** Ac3MetadataControl (p. 567)
- **Required:** False

**bitrate**

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

- **Type:** number
- **Required:** False

**lfeFilter**

When set to enabled, applies a 120Hz lowpass filter to the LFE channel prior to encoding. Only valid in codingMode32Lfe mode.

- **Type:** Ac3LfeFilter (p. 567)
- **Required:** False

**bitstreamMode**

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

- **Type:** Ac3BitstreamMode (p. 567)
**Required**: False

**AccessDenied**

**message**

*Type*: string  
*Required*: False

**AfdSignaling**

Afd Signaling

- AUTO
- FIXED
- NONE

**ArchiveContainerSettings**

Archive Container Settings

**m2tsSettings**

*Type*: M2tsSettings (p. 638)  
*Required*: False

**ArchiveGroupSettings**

Archive Group Settings

**destination**

A directory and base filename where archive files should be written.

*Type*: OutputLocationRef (p. 656)  
*Required*: True

**rolloverInterval**

Number of seconds to write to archive file before closing and starting a new one.

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

**ArchiveOutputSettings**

Archive Output Settings

**extension**

Output file extension. If excluded, this will be auto-selected from the container type.
Type: string
Required: False

**containerSettings**
Settings specific to the container type of the file.

  Type: [ArchiveContainerSettings](p. 569)
  Required: True

**nameModifier**
String concatenated to the end of the destination filename. Required for multiple outputs of the same type.

  Type: string
  Required: False

**AribDestinationSettings**
Arib Destination Settings

**AribSourceSettings**
Arib Source Settings

**AudioChannelMapping**
Audio Channel Mapping

**outputChannel**
The index of the output channel being produced.

  Type: integer
  Required: True
  Minimum: 0
  Maximum: 7

**inputChannelLevels**
Indices and gain values for each input channel that should be remixed into this output channel.

  Type: Array of type [InputChannelLevel](p. 629)
  Required: True

**AudioCodecSettings**
Audio Codec Settings

**aacSettings**

  Type: [AacSettings](p. 565)
  Required: False
ac3Settings
  Type: Ac3Settings (p. 567)
  Required: False

eac3Settings
  Type: Eac3Settings (p. 596)
  Required: False

passThroughSettings
  Type: PassThroughSettings (p. 657)
  Required: False

mp2Settings
  Type: Mp2Settings (p. 649)
  Required: False

AudioDescription

Audio Description

audioTypeControl
Determines how audio type is determined. followInput: If the input contains an ISO 639 audioType, then that value is passed through to the output. If the input contains no ISO 639 audioType, the value in Audio Type is included in the output. useConfigured: The value in Audio Type is included in the output. Note that this field and audioType are both ignored if inputType is broadcasterMixedAd.
  Type: AudioDescriptionAudioTypeControl (p. 572)
  Required: False

languageCodeControl
Choosing followInput will cause the ISO 639 language code of the output to follow the ISO 639 language code of the input. The languageCode will be used when useConfigured is set, or when followInput is selected but there is no ISO 639 language code specified by the input.
  Type: AudioDescriptionLanguageCodeControl (p. 573)
  Required: False

remixSettings
Settings that control how input audio channels are remixed into the output audio channels.
  Type: RemixSettings (p. 657)
  Required: False

audioType
Applies only if audioTypeControl is useConfigured. The values for audioType are defined in ISO-IEC 13818-1.
Properties

**Type**: AudioType (p. 576)
**Required**: False

**name**

The name of this AudioDescription. Outputs will use this name to uniquely identify this AudioDescription. Description names should be unique within this Live Event.

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

**codecSettings**

Audio codec settings.

**Type**: AudioCodecSettings (p. 570)
**Required**: False

**languageCode**

Indicates the language of the audio output track. Only used if languageControlMode is useConfigured, or there is no ISO 639 language code specified in 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).

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

**audioNormalizationSettings**

Advanced audio normalization settings.

**Type**: AudioNormalizationSettings (p. 574)
**Required**: False

**audioSelectorName**

The name of the AudioSelector used as the source for this AudioDescription.

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

**AudioDescriptionAudioTypeControl**

Audio Description Audio Type Control
FOLLOW_INPUT
USE_CONFIGURED

AudioDescriptionLanguageCodeControl
Audio Description Language Code Control
FOLLOW_INPUT
USE_CONFIGURED

AudioLanguageSelection
Audio Language Selection

languageSelectionPolicy
When set to "strict", the transport stream demux strictly identifies audio streams by their language descriptor. If a PMT update occurs such that an audio stream matching the initially selected language is no longer present then mute will be encoded until the language returns. If "loose", then on a PMT update the demux will choose another audio stream in the program with the same stream type if it can't find one with the same language.

Type: AudioLanguageSelectionPolicy (p. 573)
Required: False

languageCode
Selects a specific three-letter language code from within an audio source.

Type: string
Required: True

AudioLanguageSelectionPolicy
Audio Language Selection Policy

LOOSE
STRICT

AudioNormalizationAlgorithm
Audio Normalization Algorithm

ITU_1770_1
ITU_1770_2

AudioNormalizationAlgorithmControl
Audio Normalization Algorithm Control

CORRECT_AUDIO
AudioNormalizationSettings

Audio Normalization Settings

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
Minimum: -59.0
Maximum: 0.0

algorithmControl

When set to correctAudio the output audio is corrected using the chosen algorithm. If set to measureOnly, the audio will be measured but not adjusted.

Type: AudioNormalizationAlgorithmControl (p. 573)
Required: False

algorithm

Audio normalization algorithm to use. itu17701 conforms to the CALM Act specification, itu17702 conforms to the EBU R-128 specification.

Type: AudioNormalizationAlgorithm (p. 573)
Required: False

AudioOnlyHlsSettings

Audio Only Hls Settings

audioTrackType

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=NO 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

Type: AudioOnlyHlsTrackType (p. 575)
Required: False

audioGroupId

Specifies the group to which the audio Rendition belongs.

Type: string
Properties

**Required:** False

**audioOnlyImage**

For use with an audio only Stream. Must be a .jpg or .png file. If given, this image will be used as the cover-art for the audio only output. Ideally, it should be formatted for an iPhone screen for two reasons. The iPhone does not resize the image, it crops a centered image on the top/bottom and left/right. Additionally, this image file gets saved bit-for-bit into every 10-second segment file, so will increase bandwidth by \( \text{image file size} \times \text{segment count} \times \text{user count}. \)

*Type:* `InputLocation (p. 630)`
*Required:* False

**AudioOnlyHlsTrackType**

Audio Only Hls Track Type

- ALTERNATE_AUDIO_AUTO_SELECT
- ALTERNATE_AUDIO_AUTO_SELECT_DEFAULT
- ALTERNATE_AUDIO_NOT_AUTO_SELECT
- AUDIO_ONLY_VARIANT_STREAM

**AudioPidSelection**

Audio Pid Selection

**pid**

Selects a specific PID from within a source.

*Type:* integer
*Required:* True
*Minimum:* 0
*Maximum:* 8191

**AudioSelector**

Audio Selector

**name**

The name of this AudioSelector. AudioDescriptions will use this name to uniquely identify this Selector. Selector names should be unique per input.

*Type:* string
*Required:* True
*MinLength:* 1

**selectorSettings**

The audio selector settings.

*Type:* `AudioSelectorSettings (p. 576)`
*Required:* False
AudioSelectorSettings

Audio Selector Settings

audioLanguageSelection

  Type: AudioLanguageSelection (p. 573)
  Required: False

audioPidSelection

  Type: AudioPidSelection (p. 575)
  Required: False

AudioType

Audio Type

  CLEAN_EFFECTS
  HEARING_IMPAIRED
  UNDEFINED
  VISUAL_IMPAIRED_COMMENTARY

AuthenticationScheme

Authentication Scheme

  AKAMAI
  COMMON

AvailBlanking

Avail Blanking

state

When set to enabled, causes video, audio and captions to be blanked when insertion metadata is added.

  Type: AvailBlankingState (p. 576)
  Required: False

availBlankingImage

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

  Type: InputLocation (p. 630)
  Required: False

AvailBlankingState

Avail Blanking State
AvailConfiguration

Avail Configuration

availSettings

Ad avail settings.

  Type: AvailSettings (p. 577)
  Required: False

AvailSettings

Avail Settings

scte35TimeSignalApos

  Type: Scte35TimeSignalApos (p. 662)
  Required: False

scte35SpliceInsert

  Type: Scte35SpliceInsert (p. 661)
  Required: False

BadGatewayException

message

  Type: string
  Required: False

BlackoutSlate

Blackout Slate

networkEndBlackoutImage

Path to local file to use as Network End Blackout image. Image will be scaled to fill the entire output raster.

  Type: InputLocation (p. 630)
  Required: False

networkEndBlackout

Setting to enabled causes the encoder to blackout the video, audio, and captions, and raise the "Network Blackout Image" slate when an SCTE104/35 Network End Segmentation Descriptor is encountered.
The blackout will be lifted when the Network Start Segmentation Descriptor is encountered. The Network End and Network Start descriptors must contain a network ID that matches the value entered in "Network ID".

**Properties**

**networkId**

Provides Network ID that matches EIDR ID format (e.g., "10.XXXX/XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-C").

- **Type**: string
- **Required**: False
- **MinLength**: 34
- **MaxLength**: 34

**state**

When set to enabled, causes video, audio and captions to be blanked when indicated by program metadata.

- **Type**: BlackoutSlateState (p. 578)
- **Required**: False

**blackoutSlateImage**

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

- **Type**: InputLocation (p. 630)
- **Required**: False

---

**BlackoutSlateNetworkEndBlackout**

Blackout Slate Network End Blackout

- **DISABLED**
- **ENABLED**

**BlackoutSlateState**

Blackout Slate State

- **DISABLED**
- **ENABLED**

**BurnInAlignment**

Burn In Alignment

- **CENTERED**
- **LEFT**
- **SMART**
BurnInBackgroundColor

Burn In Background Color

BLACK
NONE
WHITE

BurnInDestinationSettings

Burn In Destination Settings

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 xPosition is provided, the horizontal caption position will be determined by the alignment parameter. All burn-in and DVB-Sub font settings must match.

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

backgroundColor

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

- **Type:** BurnInBackgroundColor (p. 579)
- **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 yPosition is provided, the caption will be positioned towards the bottom of the output. All burn-in and DVB-Sub font settings must match.

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

teletextGridControl

Controls whether a fixed grid size will be used to generate the output subtitles bitmap. Only applicable for Teletext inputs and DVB-Sub/Burn-in outputs.

- **Type:** BurnInTeletextGridControl (p. 582)
- **Required:** False

backgroundOpacity

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

**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:** 0
Maximum: 10

outlineColor
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.

Type: BurnInOutlineColor (p. 582)
Required: False

fontSize
When set to 'auto' fontSize will scale depending on the size of the output. Giving a positive integer will specify the exact font size in points. All burn-in and DVB-Sub font settings must match.

Type: string
Required: False

alignment
If no explicit xPosition or yPosition 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. Selecting "smart" justification will left-justify live subtitles and center-justify pre-recorded subtitles. All burn-in and DVB-Sub font settings must match.

Type: BurnInAlignment (p. 578)
Required: False

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

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

Type: BurnInShadowColor (p. 582)
Required: False

fontColor
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.

Type: BurnInFontColor (p. 582)
Required: False
**font**

External font file used for caption burn-in. File extension must be 'ttf' or 'tte'. Although the user can select output fonts for many different types of input captions, embedded, STL and teletext sources use a strict grid system. Using external fonts with these caption sources could cause unexpected display of proportional fonts. All burn-in and DVB-Sub font settings must match.

*Type:* InputLocation (p. 630)

*Required:* False

**BurnInFontColor**

Burn In Font Color

BLACK
BLUE
GREEN
RED
WHITE
YELLOW

**BurnInOutlineColor**

Burn In Outline Color

BLACK
BLUE
GREEN
RED
WHITE
YELLOW

**BurnInShadowColor**

Burn In Shadow Color

BLACK
NONE
WHITE

**BurnInTeletextGridControl**

Burn In Teletext Grid Control

FIXED
SCALED

**CaptionDescription**

Output groups for this Live Event. Output groups contain information about where streams should be distributed.
captionSelectorName
Specifies which input caption selector to use as a caption source when generating output captions. This field should match a captionSelector name.

Type: string
Required: True

languageDescription
Human readable information to indicate captions available for players (eg. English, or Spanish).

Type: string
Required: False

name
Name of the caption description. Used to associate a caption description with an output. Names must be unique within an event.

Type: string
Required: True

languageCode

Type: string
Required: False

destinationSettings
Additional settings for captions destination that depend on the destination type.

Type: CaptionDestinationSettings (p. 583)
Required: False

CaptionDestinationSettings
Caption Destination Settings

burnInDestinationSettings

Type: BurnInDestinationSettings (p. 579)
Required: False

scte27DestinationSettings

Type: Scte27DestinationSettings (p. 660)
Required: False

teletextDestinationSettings

Type: TeletextDestinationSettings (p. 664)
Required: False

ttmlDestinationSettings
  Type: TtmlDestinationSettings (p. 665)
  Required: False

smpteTtDestinationSettings
  Type: SmpteTtDestinationSettings (p. 664)
  Required: False

webvttDestinationSettings
  Type: WebvttDestinationSettings (p. 671)
  Required: False

embeddedPlusScte20DestinationSettings
  Type: EmbeddedPlusScte20DestinationSettings (p. 600)
  Required: False

dvbSubDestinationSettings
  Type: DvbSubDestinationSettings (p. 591)
  Required: False

embeddedDestinationSettings
  Type: EmbeddedDestinationSettings (p. 600)
  Required: False

rtmpCaptionInfoDestinationSettings
  Type: RtmpCaptionInfoDestinationSettings (p. 658)
  Required: False

aribDestinationSettings
  Type: AribDestinationSettings (p. 570)
  Required: False

scte20PlusEmbeddedDestinationSettings
  Type: Scte20PlusEmbeddedDestinationSettings (p. 660)
  Required: False

CaptionLanguageMapping
Maps a caption channel to an ISO 693-2 language code (http://www.loc.gov/standards/iso639-2), with an optional description.
languageDescription
Textual description of language

Type: string
Required: True
MinLength: 1

captionChannel
The closed caption channel being described by this CaptionLanguageMapping. Each channel mapping must have a unique channel number (maximum of 4)

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

languageCode
Three character ISO 639-2 language code (see http://www.loc.gov/standards/iso639-2)

Type: string
Required: True
MinLength: 3
MaxLength: 3

CaptionSelector
Output groups for this Live Event. Output groups contain information about where streams should be distributed.

name
Name identifier for a caption selector. This name is used to associate this caption selector with one or more caption descriptions. Names must be unique within an event.

Type: string
Required: True
MinLength: 1

languageCode
When specified this field indicates the three letter language code of the caption track to extract from the source.

Type: string
Required: False

selectorSettings
Caption selector settings.

Type: CaptionSelectorSettings (p. 586)
Required: False

CaptionSelectorSettings

Caption Selector Settings

embeddedSourceSettings

Type: EmbeddedSourceSettings (p. 600)
Required: False

scte20SourceSettings

Type: Scte20SourceSettings (p. 660)
Required: False

dvbSubSourceSettings

Type: DvbSubSourceSettings (p. 594)
Required: False

teletextSourceSettings

Type: TeletextSourceSettings (p. 664)
Required: False

aribSourceSettings

Type: AribSourceSettings (p. 570)
Required: False

scte27SourceSettings

Type: Scte27SourceSettings (p. 660)
Required: False

Channel

inputAttachments

List of input attachments for channel.

Type: Array of type InputAttachment (p. 629)
Required: False

destinations

A list of destinations of the channel. For UDP outputs, there is one destination per output. For other types (HLS, for example), there is one destination per packager.

Type: Array of type OutputDestination (p. 653)
Properties

Required: False

encoderSettings

Type: EncoderSettings (p. 601)
Required: False

egressEndpoints

The endpoints where outgoing connections initiate from

Type: Array of type ChannelEgressEndpoint (p. 588)
Required: False

inputSpecification

Type: InputSpecification (p. 634)
Required: False

channelClass

The class for this channel. STANDARD for a channel with two pipelines or SINGLE_PIPELINE for a channel with one pipeline.

Type: ChannelClass (p. 588)
Required: False

tags

A collection of key-value pairs.

Type: Tags (p. 664)
Required: False

logLevel

The log level being written to CloudWatch Logs.

Type: LogLevel (p. 635)
Required: False

roleArn

The Amazon Resource Name (ARN) of the role assumed when running the Channel.

Type: string
Required: False

name

The name of the channel. (user-mutable)

Type: string
Required: False
id
The unique id of the channel.
   Type: string
   Required: False

state
   Type: ChannelState (p. 588)
   Required: False

pipelinesRunningCount
The number of currently healthy pipelines.
   Type: integer
   Required: False

arn
The unique arn of the channel.
   Type: string
   Required: False

ChannelClass
A standard channel has two encoding pipelines and a single pipeline channel only has one.
   STANDARD
   SINGLE_PIPELINE

ChannelEgressEndpoint
sourceIp
Public IP of where a channel's output comes from
   Type: string
   Required: False

ChannelState
CREATING
CREATE_FAILED
IDLE
STARTING
RUNNING
RECOVERING
STOPPING
DELETING
DELETED
DvbNitSettings

DVB Network Information Table (NIT)

**networkName**

The network name text placed in the networkNameDescriptor inside the Network Information Table. Maximum length is 256 characters.

- **Type:** string
- **Required:** True
- **MinLength:** 1
- **MaxLength:** 256

**networkId**

The numeric value placed in the Network Information Table (NIT).

- **Type:** integer
- **Required:** True
- **Minimum:** 0
- **Maximum:** 65536

**repInterval**

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

- **Type:** integer
- **Required:** False
- **Minimum:** 25
- **Maximum:** 10000

DvbSdtOutputSdt

Dvb Sdt Output Sdt

- SDT_FOLLOW
- SDT_FOLLOW_IF_PRESENT
- SDT_MANUAL
- SDT_NONE

DvbSdtSettings

DVB Service Description Table (SDT)

**serviceName**

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

- **Type:** string
**serviceProviderName**

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

- **Type**: string
- **Required**: False
- **MinLength**: 1
- **MaxLength**: 256

**repInterval**

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

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

**outputSdt**

Selects method of inserting SDT information into output stream. The sdtFollow setting copies SDT information from input stream to output stream. The sdtFollowIfPresent setting 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. The sdtManual setting means user will enter the SDT information. The sdtNone setting means output stream will not contain SDT information.

- **Type**: DvbSdtOutputSdt (p. 589)
- **Required**: False

**DvbSubDestinationAlignment**

Dvb Sub Destination Alignment

- CENTERED
- LEFT
- SMART

**DvbSubDestinationBackgroundColor**

Dvb Sub Destination Background Color

- BLACK
- NONE
- WHITE

**DvbSubDestinationFontColor**

Dvb Sub Destination Font Color
DvbSubDestinationOutlineColor

Dvb Sub Destination Outline Color

BLACK
BLUE
GREEN
RED
WHITE
YELLOW

DvbSubDestinationSettings

Dvb Sub Destination Settings

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 xPosition 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

backgroundColor

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

Type: DvbSubDestinationBackgroundColor (p. 590)
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 yPosition 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
teletextGridControl

Controls whether a fixed grid size will be used to generate the output subtitles bitmap. Only applicable for Teletext inputs and DVB-Sub/Burn-in outputs.

Type: DvbSubDestinationTeletextGridControl (p. 594)
Required: False

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: False
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
**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:** 0
- **Maximum:** 10

**outlineColor**

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.

- **Type:** DvbSubDestinationOutlineColor (p. 591)
- **Required:** False

**fontSize**

When set to auto fontSize will scale depending on the size of the output. Giving a positive integer will specify the exact font size in points. All burn-in and DVB-Sub font settings must match.

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

**alignment**

If no explicit xPosition or yPosition 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. Selecting "smart" justification will left-justify live subtitles and center-justify pre-recorded subtitles. This option is not valid for source captions that are STL or 608/embedded. These source settings are already pre-defined by the caption stream. All burn-in and DVB-Sub font settings must match.

- **Type:** DvbSubDestinationAlignment (p. 590)
- **Required:** False

**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

**shadowColor**

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

- **Type:** DvbSubDestinationShadowColor (p. 594)
- **Required:** False
Properties

**fontColor**

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.

*Type: DvbSubDestinationFontColor (p. 590)*

*Required: False*

**font**

External font file used for caption burn-in. File extension must be 'ttf' or 'tte'. Although the user can select output fonts for many different types of input captions, embedded, STL and teletext sources use a strict grid system. Using external fonts with these caption sources could cause unexpected display of proportional fonts. All burn-in and DVB-Sub font settings must match.

*Type: InputLocation (p. 630)*

*Required: False*

**DvbSubDestinationShadowColor**

Dvb Sub Destination Shadow Color

- BLACK
- NONE
- WHITE

**DvbSubDestinationTeletextGridControl**

Dvb Sub Destination Teletext Grid Control

- FIXED
- SCALED

**DvbSubSourceSettings**

Dvb Sub Source Settings

**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*

**DvbTdtSettings**

DVB Time and Date Table (SDT)

**repInterval**

The number of milliseconds between instances of this table in the output transport stream.
Type: integer
Required: False
Minimum: 1000
Maximum: 30000

Eac3AttenuationControl
Eac3 Attenuation Control
ATTENUATE_3_DB
NONE

Eac3BitstreamMode
Eac3 Bitstream Mode
COMMENTARY
COMPLETE_MAIN
EMERGENCY
HEARING_IMPAIRED
VISUALLY_IMPAIRED

Eac3CodingMode
Eac3 Coding Mode
CODING_MODE_1_0
CODING_MODE_2_0
CODING_MODE_3_2

Eac3DcFilter
Eac3 Dc Filter
DISABLED
ENABLED

Eac3DrcLine
Eac3 Drc Line
FILM_LIGHT
FILM_STANDARD
MUSIC_LIGHT
MUSIC_STANDARD
NONE
SPEECH

Eac3DrcRf
Eac3 Drc Rf
FILM_LIGHT
FILM_STANDARD
MUSIC_LIGHT
MUSIC_STANDARD
NONE
SPEECH

**Eac3LfeControl**

Eac3 Lfe Control

LFE
NO_LFE

**Eac3LfeFilter**

Eac3 Lfe Filter

DISABLED
ENABLED

**Eac3MetadataControl**

Eac3 Metadata Control

FOLLOW_INPUT
USE_CONFIGURED

**Eac3PassthroughControl**

Eac3 Passthrough Control

NO_PASSTHROUGH
WHEN_POSSIBLE

**Eac3PhaseControl**

Eac3 Phase Control

NO_SHIFT
SHIFT_90_DEGREES

**Eac3Settings**

Eac3 Settings

**dialnorm**

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

Type: integer
### Properties

**Required**: False  
**Minimum**: 1  
**Maximum**: 31

**passthroughControl**
When set to `whenPossible`, 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.

**Type**: Eac3PassthroughControl (p. 596)  
**Required**: False

**drcLine**
Sets the Dolby dynamic range compression profile.

**Type**: Eac3DrcLine (p. 595)  
**Required**: False

**metadataControl**
When set to `followInput`, 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.

**Type**: Eac3MetadataControl (p. 596)  
**Required**: False

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

**Type**: number  
**Required**: False

**ltRtSurroundMixLevel**
Left total/Right total surround mix level. Only used for 3/2 coding mode.

**Type**: number  
**Required**: False

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

**Type**: Eac3SurroundExMode (p. 599)  
**Required**: False

**lfeControl**
When encoding 3/2 audio, setting to `lfe` enables the LFE channel
**Properties**

**Type**: Eac3LfeControl (p. 596)  
**Required**: False

**codingMode**

Dolby Digital Plus coding mode. Determines number of channels.

**Type**: Eac3CodingMode (p. 595)  
**Required**: False

**surroundMode**

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

**Type**: Eac3SurroundMode (p. 600)  
**Required**: False

**attenuationControl**

When set to attenuate3Db, applies a 3 dB attenuation to the surround channels. Only used for 3/2 coding mode.

**Type**: Eac3AttenuationControl (p. 595)  
**Required**: False

**lfeFilter**

When set to enabled, applies a 120Hz lowpass filter to the LFE channel prior to encoding. Only valid with codingMode32 coding mode.

**Type**: Eac3LfeFilter (p. 596)  
**Required**: False

**dcFilter**

When set to enabled, activates a DC highpass filter for all input channels.

**Type**: Eac3DcFilter (p. 595)  
**Required**: False

**ltRtCenterMixLevel**

Left total/Right total center mix level. Only used for 3/2 coding mode.

**Type**: number  
**Required**: False

**phaseControl**

When set to shift90Degrees, applies a 90-degree phase shift to the surround channels. Only used for 3/2 coding mode.

**Type**: Eac3PhaseControl (p. 596)
**Properties**

**bitstreamMode**

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

- **Type:** `Eac3BitstreamMode (p. 595)`
- **Required:** False

**stereoDownmix**

Stereo downmix preference. Only used for 3/2 coding mode.

- **Type:** `Eac3StereoDownmix (p. 599)`
- **Required:** False

**loRoSurroundMixLevel**

Left only/Right only surround mix level. Only used for 3/2 coding mode.

- **Type:** number
- **Required:** False

**drcRf**

Sets the profile for heavy Dolby dynamic range compression, ensures that the instantaneous signal peaks do not exceed specified levels.

- **Type:** `Eac3DrcRf (p. 595)`
- **Required:** False

**loRoCenterMixLevel**

Left only/Right only center mix level. Only used for 3/2 coding mode.

- **Type:** number
- **Required:** False

**Eac3StereoDownmix**

Eac3 Stereo Downmix

- DPL2
- LO_RO
- LT_RT
- NOT_INDICATED

**Eac3SurroundExMode**

Eac3 Surround Ex Mode

- DISABLED
Eac3SurroundMode

Eac3 Surround Mode

  - DISABLED
  - ENABLED
  - NOT_INDICATED

EmbeddedConvert608To708

Embedded Convert608 To708

  - DISABLED
  - UPCONVERT

EmbeddedDestinationSettings

Embedded Destination Settings

EmbeddedPlusScte20DestinationSettings

Embedded Plus Scte20 Destination Settings

EmbeddedScte20Detection

Embedded Scte20 Detection

  - AUTO
  - OFF

EmbeddedSourceSettings

Embedded Source Settings

scte20Detection

Set to "auto" to handle streams with intermittent and/or non-aligned SCTE-20 and Embedded captions.

  - Type: EmbeddedScte20Detection (p. 600)
  - Required: False

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**

If 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.

- **Type**: EmbeddedConvert608To708 (p. 600)
- **Required**: False

**source608TrackNumber**

This field is unused and deprecated.

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 5

**EncoderSettings**

Encoder Settings

**timecodeConfig**

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

- **Type**: TimecodeConfig (p. 665)
- **Required**: True

**outputGroups**

- **Type**: Array of type OutputGroup (p. 654)
- **Required**: True

**audioDescriptions**

- **Type**: Array of type AudioDescription (p. 571)
- **Required**: True

**availConfiguration**

Event-wide configuration settings for ad avail insertion.

- **Type**: AvailConfiguration (p. 577)
- **Required**: False

**captionDescriptions**

Settings for caption descriptions

- **Type**: Array of type CaptionDescription (p. 582)
- **Required**: False
globalConfiguration

Configuration settings that apply to the event as a whole.

Type: GlobalConfiguration (p. 604)
Required: False

videoDescriptions

Type: Array of type VideoDescription (p. 667)
Required: True

availBlanking

Settings for ad avail blanking.

Type: AvailBlanking (p. 576)
Required: False

blackoutSlate

Settings for blackout slate.

Type: BlackoutSlate (p. 577)
Required: False

FecOutputIncludeFec

Fec Output Include Fec

- COLUMN
- COLUMN_AND_ROW

FecOutputSettings

Fec Output Settings

rowLength

Parameter L from SMPTE 2022-1. The width of the FEC protection matrix. Must be between 1 and 20, inclusive. If only Column FEC is used, then larger values increase robustness. If Row FEC is used, then this is the number of transport stream packets per row error correction packet, and the value must be between 4 and 20, inclusive, if includeFec is columnAndRow. If includeFec is column, this value must be 1 to 20, inclusive.

Type: integer
Required: False
Minimum: 1
Maximum: 20

columnDepth

Parameter D from SMPTE 2022-1. The height of the FEC protection matrix. The number of transport stream packets per column error correction packet. Must be between 4 and 20, inclusive.
Properties

Type: integer  
Required: False  
Minimum: 4  
Maximum: 20

**includeFec**

Enables column only or column and row based FEC  
Type: **FecOutputIncludeFec (p. 602)**  
Required: False

**FixedAfd**

Fixed Afd  
AFD_0000  
AFD_0010  
AFD_0011  
AFD_0100  
AFD_1000  
AFD_1001  
AFD_1010  
AFD_1011  
AFD_1101  
AFD_1110  
AFD_1111

**FrameCaptureGroupSettings**

Frame Capture Group Settings  
**destination**

The destination for the frame capture files. Either the URI for an Amazon S3 bucket and object, plus a file name prefix (for example, s3ssl://sportsDelivery/highlights/20180820/curling_) or the URI for a MediaStore container, plus a file name prefix (for example, mediastoressl://sportsDelivery/20180820/curling_). The final file names consist of the prefix from the destination field (for example, "curling_") + name modifier + the counter (5 digits, starting from 00001) + extension (which is always .jpg). For example, curlingLow.00001.jpg  
Type: **OutputLocationRef (p. 656)**  
Required: True

**FrameCaptureOutputSettings**

Frame Capture Output Settings  
**nameModifier**

Required if the output group contains more than one output. This modifier forms part of the output file name.  
Type: string
**Required**: False

**FrameCaptureSettings**
Frame Capture Settings

**captureInterval**
The frequency, in seconds, for capturing frames for inclusion in the output. For example, "10" means capture a frame every 10 seconds.

*Type*: integer  
*Required*: True  
*Minimum*: 1  
*Maximum*: 3600

**GatewayTimeoutException**

**message**

*Type*: string  
*Required*: False

**GlobalConfiguration**
Global Configuration

**inputLossBehavior**
Settings for system actions when input is lost.

*Type*: InputLossBehavior (p. 631)  
*Required*: False

**supportLowFramerateInputs**
Adjusts video input buffer for streams with very low video framerates. This is commonly set to enabled for music channels with less than one video frame per second.

*Type*: GlobalConfigurationLowFramerateInputs (p. 605)  
*Required*: False

**outputLockingMode**
Indicates how MediaLive pipelines are synchronized. PIPELINELOCKING - MediaLive will attempt to synchronize the output of each pipeline to the other. EPOCHLOCKING - MediaLive will attempt to synchronize the output of each pipeline to the Unix epoch.

*Type*: GlobalConfigurationOutputLockingMode (p. 605)  
*Required*: False

**initialAudioGain**
Value to set the initial audio gain for the Live Event.
inputEndAction

Indicates the action to take when the current input completes (e.g. end-of-file). When switchAndLoopInputs is configured the encoder will restart at the beginning of the first input. When "none" is configured the encoder will transcode either black, a solid color, or a user specified slate images per the "Input Loss Behavior" configuration until the next input switch occurs (which is controlled through the Channel Schedule API).

   Type: GlobalConfigurationInputEndAction (p. 605)
   Required: False

outputTimingSource

Indicates whether the rate of frames emitted by the Live encoder should be paced by its system clock (which optionally may be locked to another source via NTP) or should be locked to the clock of the source that is providing the input stream.

   Type: GlobalConfigurationOutputTimingSource (p. 605)
   Required: False

GlobalConfigurationInputEndAction

Global Configuration Input End Action

   NONE
   SWITCH_AND_LOOP_INPUTS

GlobalConfigurationLowFramerateInputs

Global Configuration Low Framerate Inputs

   DISABLED
   ENABLED

GlobalConfigurationOutputLockingMode

Global Configuration Output Locking Mode

   EPOCH_LOCKING
   PIPELINE_LOCKING

GlobalConfigurationOutputTimingSource

Global Configuration Output Timing Source

   INPUT_CLOCK
   SYSTEM_CLOCK
**H264AdaptiveQuantization**
H264 Adaptive Quantization
- HIGH
- HIGHER
- LOW
- MAX
- MEDIUM
- OFF

**H264ColorMetadata**
H264 Color Metadata
- IGNORE
- INSERT

**H264EntropyEncoding**
H264 Entropy Encoding
- CABAC
- CAVLC

**H264FlickerAq**
H264 Flicker Aq
- DISABLED
- ENABLED

**H264FramerateControl**
H264 Framerate Control
- INITIALIZE_FROM_SOURCE
- SPECIFIED

**H264GopBReference**
H264 Gop BReference
- DISABLED
- ENABLED

**H264GopSizeUnits**
H264 Gop Size Units
- FRAMES
- SECONDS
H264Level

H264 Level

H264_LEVEL_1
H264_LEVEL_1_1
H264_LEVEL_1_2
H264_LEVEL_1_3
H264_LEVEL_2
H264_LEVEL_2_1
H264_LEVEL_2_2
H264_LEVEL_3
H264_LEVEL_3_1
H264_LEVEL_3_2
H264_LEVEL_4
H264_LEVEL_4_1
H264_LEVEL_4_2
H264_LEVEL_5
H264_LEVEL_5_1
H264_LEVEL_5_2
H264_LEVEL.AUTO

H264LookAheadRateControl

H264 Look Ahead Rate Control

HIGH
LOW
MEDIUM

H264ParControl

H264 Par Control

INITIALIZE_FROM_SOURCE
SPECIFIED

H264Profile

H264 Profile

BASELINE
HIGH
HIGH_10BIT
HIGH_422
HIGH_422_10BIT
MAIN

H264RateControlMode

H264 Rate Control Mode
CBR
QVBR
VBR

**H264ScanType**

H264 Scan Type

INTERLACED
PROGRESSIVE

**H264SceneChangeDetect**

H264 Scene Change Detect

DISABLED
ENABLED

**H264Settings**

H264 Settings

**minInterval**

Only meaningful if sceneChangeDetect is set to enabled. 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. Note: Maximum GOP stretch = GOP size + Min-I-interval - 1

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

**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. This field is optional; when no value is specified the encoder will choose the number of slices based on encode resolution.

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

**parNumerator**

Pixel Aspect Ratio numerator.

- **Type**: integer
Required: False

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

Type: H264GopSizeUnits (p. 606)
Required: False

subgopLength
If set to fixed, use gopNumBFrames B-frames per sub-GOP. If set to dynamic, optimize the number of B-frames used for each sub-GOP to improve visual quality.

Type: H264SubGopLength (p. 614)
Required: False

maxBitrate
For QVBR: See the tooltip for Quality level For VBR: Set the maximum bitrate in order to accommodate expected spikes in the complexity of the video.

Type: integer
Required: False
Minimum: 1000

bitrate
Average bitrate in bits/second. Required when the rate control mode is VBR or CBR. Not used for QVBR. In an MS Smooth output group, each output must have a unique value when its bitrate is rounded down to the nearest multiple of 1000.

Type: integer
Required: False
Minimum: 1000

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

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

temporalAq
If set to enabled, adjust quantization within each frame based on temporal variation of content complexity.

Type: H264TemporalAq (p. 614)
Required: False
**afdSignaling**
Indicates that AFD values will be written into the output stream. If afdSignaling is "auto", the system will try to preserve the input AFD value (in cases where multiple AFD values are valid). If set to "fixed", the AFD value will be the value configured in the fixedAfd parameter.

- **Type**: AfdSignaling (p. 569)
- **Required**: False

**timecodeInsertion**
Determines how timecodes should be inserted into the video elementary stream. - 'disabled': Do not include timecodes - 'picTimingSei': Pass through picture timing SEI messages from the source specified in Timecode Config

- **Type**: H264TimecodeInsertionBehavior (p. 614)
- **Required**: False

**bufSize**
Size of buffer (HRD buffer model) in bits/second.

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

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

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

**framerateControl**
This field indicates how the output video frame rate is specified. If "specified" is selected then the output video frame rate is determined by framerateNumerator and framerateDenominator, else if "initializeFromSource" is selected then the output video frame rate will be set equal to the input video frame rate of the first input.

- **Type**: H264FramerateControl (p. 606)
- **Required**: False

**qvbrQualityLevel**
Controls the target quality for the video encode. Applies only when the rate control mode is QVBR. Set values for the QVBR quality level field and Max bitrate field that suit your most important viewing devices. Recommended values are: - Primary screen: Quality level: 8 to 10. Max bitrate: 4M - PC or tablet: Quality level: 7. Max bitrate: 1.5M to 3M - Smartphone: Quality level: 6. Max bitrate: 1M to 1.5M

- **Type**: integer
- **Required**: False
- **Minimum**: 1
- **Maximum**: 10
fixedAfd

Four bit AFD value to write on all frames of video in the output stream. Only valid when afdSignaling is set to 'Fixed'.

Type: FixedAfd (p. 603)  
Required: False

level

H.264 Level.

Type: H264Level (p. 607)  
Required: False

lookAheadRateControl

Amount of lookahead. A value of low can decrease latency and memory usage, while high can produce better quality for certain content.

Type: H264LookAheadRateControl (p. 607)  
Required: False

profile

H.264 Profile.

Type: H264Profile (p. 607)  
Required: False

framerateNumerator

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

Type: integer  
Required: False  
Minimum: 1

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

entropyEncoding

Entropy encoding mode. Use cabac (must be in Main or High profile) or cavlc.

Type: H264EntropyEncoding (p. 606)  
Required: False
framerateDenominator

Framerate denominator.

Type: integer
Required: False
Minimum: 1

spatialAq

If set to enabled, adjust quantization within each frame based on spatial variation of content complexity.

Type: H264SpatialAq (p. 614)
Required: False

adaptiveQuantization

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

Type: H264AdaptiveQuantization (p. 606)
Required: False

colorMetadata

Includes colorspace metadata in the output.

Type: H264ColorMetadata (p. 606)
Required: False

gopSize

GOP size (keyframe interval) in units of either frames or seconds per gopSizeUnits. Must be greater than zero.

Type: number
Required: False
Minimum: 1.0

numRefFrames

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

gopBReference

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

Type: H264GopBReference (p. 606)
Required: False
**parControl**

This field indicates how the output pixel aspect ratio is specified. If "specified" is selected then the output video pixel aspect ratio is determined by parNumerator and parDenominator, else if "initializeFromSource" is selected then the output pixel aspect ratio will be set equal to the input video pixel aspect ratio of the first input.

*Type:* H264ParControl (p. 607)  
*Required:* False

**parDenominator**

Pixel Aspect Ratio denominator.

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

**syntax**

Produces a bitstream compliant with SMPTE RP-2027.

*Type:* H264Syntax (p. 614)  
*Required:* False

**sceneChangeDetect**

Scene change detection. - On: inserts I-frames when scene change is detected. - Off: does not force an I-frame when scene change is detected.

*Type:* H264SceneChangeDetect (p. 608)  
*Required:* False

**scanType**

Sets the scan type of the output to progressive or top-field-first interlaced.

*Type:* H264ScanType (p. 608)  
*Required:* False

**flickerAq**

If set to enabled, adjust quantization within each frame to reduce flicker or ‘pop’ on I-frames.

*Type:* H264FlickerAq (p. 606)  
*Required:* False

**gopNumBFrames**

Number of B-frames between reference frames.

*Type:* integer  
*Required:* False  
*Minimum:* 0  
*Maximum:* 7
rateControlMode
Rate control mode. QVBR: Quality will match the specified quality level except when it is constrained by the maximum bitrate. Recommended if you or your viewers pay for bandwidth. VBR: Quality and bitrate vary, depending on the video complexity. Recommended instead of QVBR if you want to maintain a specific average bitrate over the duration of the channel. CBR: Quality varies, depending on the video complexity. Recommended only if you distribute your assets to devices that cannot handle variable bitrates.

Type: H264RateControlMode (p. 607)
Required: False

H264SpatialAq
H264 Spatial Aq
DISABLED
ENABLED

H264SubGopLength
H264 Sub Gop Length
DYNAMIC
FIXED

H264Syntax
H264 Syntax
DEFAULT
RP2027

H264TemporalAq
H264 Temporal Aq
DISABLED
ENABLED

H264TimecodeInsertionBehavior
H264 Timecode Insertion Behavior
DISABLED
PIC_TIMING_SEI

HlsAdMarkers
Hls Ad Markers
ADOBE
ELEMENTAL
ELEMENTAL_SCTE35

**HlsAkamaiHttpTransferMode**

Hls Akamai Http Transfer Mode

- CHUNKED
- NON_CHUNKED

**HlsAkamaiSettings**

Hls Akamai Settings

**salt**

Salt for authenticated Akamai.

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

**httpTransferMode**

Specify whether or not to use chunked transfer encoding to Akamai. User should contact Akamai to enable this feature.

  - **Type**: HlsAkamaiHttpTransferMode (p. 615)
  - **Required**: False

**numRetries**

Number of retry attempts that will be made before the Live Event is put into an error state.

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

**restartDelay**

If a streaming output fails, number of seconds to wait until a restart is initiated. A value of 0 means never restart.

  - **Type**: integer
  - **Required**: False
  - **Minimum**: 0
  - **Maximum**: 15

**connectionRetryInterval**

Number of seconds to wait before retrying connection to the CDN if the connection is lost.

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

**filecacheDuration**

Size in seconds of file cache for streaming outputs.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
- **Maximum:** 600

**token**

Token parameter for authenticated akamai. If not specified, _gda_ is used.

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

**HlsBasicPutSettings**

Hls Basic Put Settings

**numRetries**

Number of retry attempts that will be made before the Live Event is put into an error state.

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

**restartDelay**

If a streaming output fails, number of seconds to wait until a restart is initiated. A value of 0 means never restart.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
- **Maximum:** 15

**connectionRetryInterval**

Number of seconds to wait before retrying connection to the CDN if the connection is lost.

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

**filecacheDuration**

Size in seconds of file cache for streaming outputs.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
Maximum: 600

**HlsCaptionLanguageSetting**

Hls Caption Language Setting

- INSERT
- NONE
- OMIT

**HlsCdnSettings**

Hls Cdn Settings

- **hlsAkamaiSettings**
  - **Type:** HlsAkamaiSettings (p. 615)
  - **Required:** False

- **hlsWebdavSettings**
  - **Type:** HlsWebdavSettings (p. 628)
  - **Required:** False

- **hlsBasicPutSettings**
  - **Type:** HlsBasicPutSettings (p. 616)
  - **Required:** False

- **hlsMediaStoreSettings**
  - **Type:** HlsMediaStoreSettings (p. 625)
  - **Required:** False

**HlsClientCache**

Hls Client Cache

- DISABLED
- ENABLED

**HlsCodecSpecification**

Hls Codec Specification

- RFC_4281
- RFC_6381

**HlsDirectoryStructure**

Hls Directory Structure
SINGLE_DIRECTORY
SUBDIRECTORY_PER_STREAM

**HlsEncryptionType**

Hls Encryption Type

- AES128
- SAMPLE_AES

**HlsGroupSettings**

Hls Group Settings

**segmentsPerSubdirectory**

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

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

**ivInManifest**

For use with encryptionType. The IV (Initialization Vector) is a 128-bit number used in conjunction with the key for encrypting blocks. If set to "include", IV is listed in the manifest, otherwise the IV is not in the manifest.

- **Type**: HlsIvInManifest
- **Required**: False

**outputSelection**

MANIFESTSANDSEGMENTS: Generates manifests (master manifest, if applicable, and media manifests) for this output group. SEGMENTSONLY: Does not generate any manifests for this output group.

- **Type**: HlsOutputSelection
- **Required**: False

**destination**

A directory or HTTP destination for the HLS segments, manifest files, and encryption keys (if enabled).

- **Type**: OutputLocationRef
- **Required**: True

**encryptionType**

Encrypts the segments with the given encryption scheme. Exclude this parameter if no encryption is desired.

- **Type**: HlsEncryptionType
**Properties**

**Required**: False

**indexNSegments**
Applies only if Mode field is LIVE. Specifies the maximum number of segments in the media manifest file. After this maximum, older segments are removed from the media manifest. This number must be less than or equal to the Keep Segments field.

  * **Type**: integer
  * **Required**: False
  * **Minimum**: 3

**constantIv**
For use with encryptionType. This is a 128-bit, 16-byte hex value represented by a 32-character text string. If ivSource is set to "explicit" then this parameter is required and is used as the IV for encryption.

  * **Type**: string
  * **Required**: False
  * **MinLength**: 32
  * **MaxLength**: 32

**timedMetadataId3Frame**
Indicates ID3 frame that has the timecode.

  * **Type**: HlsTimedMetadataId3Frame (p. 627)
  * **Required**: False

**baseUrlManifest**
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

**captionLanguageSetting**
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.

  * **Type**: HlsCaptionLanguageSetting (p. 617)
  * **Required**: False

**minSegmentLength**
When set, minimumSegmentLength 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: False
Minimum: 0

mode

If "vod", all segments are indexed and kept permanently in the destination and manifest. If "live", only
the number segments specified in keepSegments and indexNSegments are kept; newer segments replace
older segments, which may prevent players from rewinding all the way to the beginning of the event.
VOD mode uses HLS EXT-X-PLAYLIST-TYPE of EVENT while the channel is running, converting it to a
"VOD" type manifest on completion of the stream.

Type: HlsMode (p. 626)
Required: False

keyProviderSettings

The key provider settings.

Type: KeyProviderSettings (p. 635)
Required: False

manifestCompression

When set to gzip, compresses HLS playlist.

Type: HlsManifestCompression (p. 625)
Required: False

ivSource

For use with encryptionType. The IV (Initialization Vector) is a 128-bit number used in conjunction with
the key for encrypting blocks. If this setting is "followsSegmentNumber", it will cause the IV to change
every segment (to match the segment number). If this is set to "explicit", you must enter a constantIV
value.

Type: HlsIvSource (p. 624)
Required: False

tsFileMode

SEGMENTEDFILES: Emit the program as segments - multiple .ts media files. SINGLEFILE: Applies only
if Mode field is VOD. Emit the program as a single .ts media file. The media manifest includes #EXT-X-
BYTE_RANGE tags to index segments for playback. A typical use for this value is when sending the output
to AWS Elemental MediaConvert, which can accept only a single media file. Playback while the channel is
running is not guaranteed due to HTTP server caching.

Type: HlsTsFileMode (p. 628)
Required: False

manifestDurationFormat

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

Type: HlsManifestDurationFormat (p. 625)
**Properties**

- **Required**: False

**keyFormatVersions**

Either a single positive integer version value or a slash delimited list of version values (1/2/3).

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

**streamInfResolution**

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

  - **Type**: HlsStreamInfResolution (p. 627)
  - **Required**: False

**timestampDeltaMilliseconds**

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

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

**baseUrlContent**

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

**segmentationMode**

useInputSegmentation has been deprecated. The configured segment size is always used.

  - **Type**: HlsSegmentationMode (p. 627)
  - **Required**: False

**captionLanguageMappings**

Mapping of up to 4 caption channels to caption languages. Is only meaningful if captionLanguageSetting is set to “insert”.

  - **Type**: Array of type CaptionLanguageMapping (p. 584)
  - **Required**: False

**clientCache**

When set to "disabled", sets the #EXT-X-ALLOW-CACHE:no tag in the manifest, which prevents clients from saving media segments for later replay.

  - **Type**: HlsClientCache (p. 617)
**Properties**

**Required:** False

**codecSpecification**

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

**Type:** HlsCodecSpecification (p. 617)

**keepSegments**

Applies only if Mode field is LIVE. Specifies the number of media segments (.ts files) to retain in the destination directory.

**Type:** integer

**Minimum:** 1

**redundantManifest**

ENABLED: The master manifest (.m3u8 file) for each pipeline includes information about both pipelines: first its own media files, then the media files of the other pipeline. This feature allows playout device that support stale manifest detection to switch from one manifest to the other, when the current manifest seems to be stale. There are still two destinations and two master manifests, but both master manifests reference the media files from both pipelines. DISABLED: The master manifest (.m3u8 file) for each pipeline includes information about its own pipeline only. For an HLS output group with MediaPackage as the destination, the DISABLED behavior is always followed. MediaPackage regenerates the manifests it serves to players so a redundant manifest from MediaLive is irrelevant.

**Type:** HlsRedundantManifest (p. 627)

**timedMetadataId3Period**

Timed Metadata interval in seconds.

**Type:** integer

**Minimum:** 0

**programDateTime**

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 timestampOffset.

**Type:** HlsProgramDateTime (p. 627)

**directoryStructure**

Place segments in subdirectories.

**Type:** HlsDirectoryStructure (p. 617)

**Required:** False
keyFormat

The value specifies how the key is represented in the resource identified by the URI. If parameter is absent, an implicit value of "identity" is used. A reverse DNS string can also be given.

Type: string
Required: False

inputLossAction

Parameter that control output group behavior on input loss.

Type: InputLossActionForHlsOut (p. 631)
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 HlsAdMarkers (p. 614)
Required: False

programDateTimePeriod

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

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

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: False
Minimum: 1

hlsCdnSettings

Parameters that control interactions with the CDN.

Type: HlsCdnSettings (p. 617)
Required: False

iFrameOnlyPlaylists

DISABLED: Do not create an I-frame-only manifest, but do create the master and media manifests (according to the Output Selection field). STANDARD: Create an I-frame-only manifest for each output that contains video, as well as the other manifests (according to the Output Selection field). The I-frame manifest contains a #EXT-X-I-FRAMES-ONLY tag to indicate it is I-frame only, and one or more #EXT-X-BYTERANGE entries identifying the I-frame position. For example, #EXT-X-BYTERANGE:160364@1461888"
**Type:** IFrameOnlyPlaylistType (p. 629)
**Required:** False

### HlsInputSettings

**Hls Input Settings**

**retries**

The number of consecutive times that attempts to read a manifest or segment must fail before the input is considered unavailable.

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

**bandwidth**

When specified the HLS stream with the m3u8 BANDWIDTH that most closely matches this value will be chosen, otherwise the highest bandwidth stream in the m3u8 will be chosen. The bitrate is specified in bits per second, as in an HLS manifest.

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

**retryInterval**

The number of seconds between retries when an attempt to read a manifest or segment fails.

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

**bufferSegments**

When specified, reading of the HLS input will begin this many buffer segments from the end (most recently written segment). When not specified, the HLS input will begin with the first segment specified in the m3u8.

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

### HlsIvInManifest

**Hls Iv In Manifest**

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

### HlsIvSource

**Hls Iv Source**
**EXPLICIT**

**FOLLOWS_SEGMENT_NUMBER**

**HlsManifestCompression**

Hls Manifest Compression

- GZIP
- NONE

**HlsManifestDurationFormat**

Hls Manifest Duration Format

- FLOATING_POINT
- INTEGER

**HlsMediaStoreSettings**

Hls Media Store Settings

**mediaStoreStorageClass**

When set to temporal, output files are stored in non-persistent memory for faster reading and writing.

- **Type:** HlsMediaStoreStorageClass (p. 626)
- **Required:** False

**numRetries**

Number of retry attempts that will be made before the Live Event is put into an error state.

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

**restartDelay**

If a streaming output fails, number of seconds to wait until a restart is initiated. A value of 0 means never restart.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
- **Maximum:** 15

**connectionRetryInterval**

Number of seconds to wait before retrying connection to the CDN if the connection is lost.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
**filecacheDuration**
Size in seconds of file cache for streaming outputs.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
- **Maximum:** 600

**HlsMediaStoreStorageClass**
Hls Media Store Storage Class

- TEMPORAL

**HlsMode**
Hls Mode

- LIVE
- VOD

**HlsOutputSelection**
Hls Output Selection

- MANIFESTS_AND_SEGMENTS
- SEGMENTS_ONLY

**HlsOutputSettings**
Hls Output Settings

**segmentModifier**
String concatenated to end of segment filenames.

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

**hlsSettings**
Settings regarding the underlying stream. These settings are different for audio-only outputs.

- **Type:** HlsSettings (p. 627)
- **Required:** True

**nameModifier**
String concatenated to the end of the destination filename. Accepts "Format Identifiers ":#formatIdentifierParameters.

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

**MinLength:** 1

**HlsProgramDateTime**

Hls Program Date Time

- EXCLUDE
- INCLUDE

**HlsRedundantManifest**

Hls Redundant Manifest

- DISABLED
- ENABLED

**HlsSegmentationMode**

Hls Segmentation Mode

- USE_INPUT_SEGMENTATION
- USE_SEGMENT_DURATIONS

**HlsSettings**

Hls Settings

- **standardHlsSettings**
  - Type: StandardHlsSettings (p. 664)
  - Required: False

- **audioOnlyHlsSettings**
  - Type: AudioOnlyHlsSettings (p. 574)
  - Required: False

**HlsStreamInfResolution**

Hls Stream Inf Resolution

- EXCLUDE
- INCLUDE

**HlsTimedMetadataId3Frame**

Hls Timed Metadata Id3 Frame

- NONE
- PRIV
- TDRL
**HlsTsFileMode**

Hls Ts File Mode

- SEGMENTED_FILES
- SINGLE_FILE

**HlsWebdavHttpTransferMode**

Hls Webdav Http Transfer Mode

- CHUNKED
- NON_CHUNKED

**HlsWebdavSettings**

Hls Webdav Settings

**httpTransferMode**

Specify whether or not to use chunked transfer encoding to WebDAV.

- **Type**: HlsWebdavHttpTransferMode (p. 628)
- **Required**: False

**numRetries**

Number of retry attempts that will be made before the Live Event is put into an error state.

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

**restartDelay**

If a streaming output fails, number of seconds to wait until a restart is initiated. A value of 0 means never restart.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 15

**connectionRetryInterval**

Number of seconds to wait before retrying connection to the CDN if the connection is lost.

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

**fileCacheDuration**

Size in seconds of file cache for streaming outputs.
Properties

Type: integer
Required: False
Minimum: 0
Maximum: 600

IFrameOnlyPlaylistType

When set to "standard", an I-Frame only playlist will be written out for each video output in the output group. This I-Frame only playlist will contain byte range offsets pointing to the I-frame(s) in each segment.

DISABLED
STANDARD

InputAttachment

inputId

The ID of the input

Type: string
Required: False

inputAttachmentName

User-specified name for the attachment. This is required if the user wants to use this input in an input switch action.

Type: string
Required: False

inputSettings

Settings of an input (caption selector, etc.)

Type: InputSettings (p. 633)
Required: False

InputChannelLevel

Input Channel Level

inputChannel

The index of the input channel used as a source.

Type: integer
Required: True
Minimum: 0
Maximum: 15

gain

Remixing value. Units are in dB and acceptable values are within the range from -60 (mute) and 6 dB.
Properties

Type: integer
Required: True
Minimum: -60
Maximum: 6

InputCodec
codec in increasing order of complexity

MPEG2
AVC
HEVC

InputDeblockFilter
Input Deblock Filter

DISABLED
ENABLED

InputDenoiseFilter
Input Denoise Filter

DISABLED
ENABLED

InputFilter
Input Filter

AUTO
DISABLED
FORCED

InputLocation
Input Location

passwordParam
key used to extract the password from EC2 Parameter store

Type: string
Required: False

uri
Uniform Resource Identifier - This should be a path to a file accessible to the Live system (eg. a http:// URI) depending on the output type. For example, a RTMP destination should have a uri similar to: “rtmp://fmsserver/live”. 
**InputLossActionForHlsOut**

Input Loss Action For Hls Out

- EMIT_OUTPUT
- PAUSE_OUTPUT

**InputLossActionForMsSmoothOut**

Input Loss Action For Ms Smooth Out

- EMIT_OUTPUT
- PAUSE_OUTPUT

**InputLossActionForRtmpOut**

Input Loss Action For Rtmp Out

- EMIT_OUTPUT
- PAUSE_OUTPUT

**InputLossActionForUdpOut**

Input Loss Action For Udp Out

- DROP_PROGRAM
- DROP_TS
- EMIT_PROGRAM

**InputLossBehavior**

Input Loss Behavior

**inputLossImageType**

Indicates whether to substitute a solid color or a slate into the output after input loss exceeds blackFrameMsec.

*Type: InputLossImageType (p. 632)*
Properties

**inputLossImageColor**
When input loss image type is "color" this field specifies the color to use. Value: 6 hex characters representing the values of RGB.

- **Type**: string
- **Required**: False
- **MinLength**: 6
- **MaxLength**: 6

**inputLossImageSlate**
When input loss image type is "slate" these fields specify the parameters for accessing the slate.

- **Type**: InputLocation (p. 630)
- **Required**: False

**repeatFrameMsec**
On input loss, the number of milliseconds to repeat the previous picture before substituting black into the output. A value x, where 0 <= x <= 1,000,000 and a value of 1,000,000 will be interpreted as infinite.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 1000000

**blackFrameMsec**
On input loss, the number of milliseconds to substitute black into the output before switching to the frame specified by inputLossImageType. A value x, where 0 <= x <= 1,000,000 and a value of 1,000,000 will be interpreted as infinite.

- **Type**: integer
- **Required**: False
- **Minimum**: 0
- **Maximum**: 1000000

**InputLossImageType**
Input Loss Image Type

- COLOR
- SLATE

**InputMaximumBitrate**
Maximum input bitrate in megabits per second. Bitrates up to 50 Mbps are supported currently.

- MAX_10_MBPS
- MAX_20_MBPS
- MAX_50_MBPS
InputResolution
Input resolution based on lines of vertical resolution in the input; SD is less than 720 lines, HD is 720 to 1080 lines, UHD is greater than 1080 lines
- SD
- HD
- UHD

InputSettings
Live Event input parameters. There can be multiple inputs in a single Live Event.

sourceEndBehavior
Loop input if it is a file. This allows a file input to be streamed indefinitely.
- Type: InputSourceEndBehavior (p. 634)
- Required: False

deblockFilter
Enable or disable the deblock filter when filtering.
- Type: InputDeblockFilter (p. 630)
- Required: False

audioSelectors
Used to select the audio stream to decode for inputs that have multiple available.
- Type: Array of type AudioSelector (p. 575)
- Required: False

networkInputSettings
Input settings.
- Type: NetworkInputSettings (p. 652)
- Required: False

inputFilter
Turns on the filter for this input. MPEG-2 inputs have the deblocking filter enabled by default. 1) auto - filtering will be applied depending on input type/quality 2) disabled - no filtering will be applied to the input 3) forced - filtering will be applied regardless of input type
- Type: InputFilter (p. 630)
- Required: False

videoSelector
Informs which video elementary stream to decode for input types that have multiple available.
- Type: VideoSelector (p. 669)
Required: False

**filterStrength**

Adjusts the magnitude of filtering from 1 (minimal) to 5 (strongest).

- **Type:** integer
- **Required:** False
- **Minimum:** 1
- **Maximum:** 5

**denoiseFilter**

Enable or disable the denoise filter when filtering.

- **Type:** InputDenoiseFilter (p. 630)
- **Required:** False

**captionSelectors**

Used to select the caption input to use for inputs that have multiple available.

- **Type:** Array of type CaptionSelector (p. 585)
- **Required:** False

**InputSourceEndBehavior**

Input Source End Behavior

- **CONTINUE**
- **LOOP**

**InputSpecification**

**codec**

Input codec

- **Type:** InputCodec (p. 630)
- **Required:** False

**resolution**

Input resolution, categorized coarsely

- **Type:** InputResolution (p. 633)
- **Required:** False

**maximumBitrate**

Maximum input bitrate, categorized coarsely

- **Type:** InputMaximumBitrate (p. 632)
- **Required:** False
**InternalServiceError**

message

Type: string  
Required: False

**InvalidRequest**

message

Type: string  
Required: False

**KeyProviderSettings**

Key Provider Settings

**staticKeySettings**

Type: StaticKeySettings (p. 664)  
Required: False

**LimitExceeded**

message

Type: string  
Required: False

**LogLevel**

The log level the user wants for their channel.

ERROR  
WARNING  
INFO  
DEBUG  
DISABLED

**M2tsAbsentInputAudioBehavior**

M2ts Absent Input Audio Behavior

DROP  
ENCODE_SILENCE

**M2tsArib**

M2ts Arib
DISABLED
ENABLED

**M2tsAribCaptionsPidControl**
M2ts Arib Captions Pid Control
AUTO
USE_CONFIGURED

**M2tsAudioBufferModel**
M2ts Audio Buffer Model
ATSC
DVB

**M2tsAudioInterval**
M2ts Audio Interval
VIDEO_AND_FIXED_INTERVALS
VIDEO_INTERVAL

**M2tsAudioStreamType**
M2ts Audio Stream Type
ATSC
DVB

**M2tsBufferModel**
M2ts Buffer Model
MULTIPLEX
NONE

**M2tsCcDescriptor**
M2ts Cc Descriptor
DISABLED
ENABLED

**M2tsEbifControl**
M2ts Ebif Control
NONE
PASSTHROUGH
M2tsEbpPlacement

M2ts Ebp Placement

VIDEO_AND_AUDIO_PIDS
VIDEO_PID

M2tsEsRateInPes

M2ts Es Rate In Pes

EXCLUDE
INCLUDE

M2tsKlv

M2ts Klv

NONE
PASSTHROUGH

M2tsPcrControl

M2ts Pcr Control

CONFIGURED_PCR_PERIOD
PCR_EVERY_PES_PACKET

M2tsRateMode

M2ts Rate Mode

CBR
VBR

M2tsScte35Control

M2ts Scte35 Control

NONE
PASSTHROUGH

M2tsSegmentationMarkers

M2ts Segmentation Markers

EBP
EBP_LEGACY
NONE
PSI_SEGSTART
RAI_ADAPT
RAI_SEGSTART
**M2tsSegmentationStyle**

M2ts Segmentation Style

- MAINTAIN_CADENCE
- RESET_CADENCE

**M2tsSettings**

M2ts Settings

**audioStreamType**

When set to atsc, uses stream type = 0x81 for AC3 and stream type = 0x87 for EAC3. When set to dvb, uses stream type = 0x06.

- **Type:** M2tsAudioStreamType (p. 636)
- **Required:** False

**ecmPid**

This field is unused and deprecated.

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

**dvbTeletextPid**

Packet Identifier (PID) for input source DVB Teletext data to this output. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20).8182 (or 0x1ff6).

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

**aribCaptionsPidControl**

If set to auto, pid number used for ARIB Captions will be auto-selected from unused pids. If set to useConfigured, ARIB Captions will be on the configured pid number.

- **Type:** M2tsAribCaptionsPidControl (p. 636)
- **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.

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

**rateMode**

When vbr, does not insert null packets into transport stream to fill specified bitrate. The bitrate setting acts as the maximum bitrate when vbr is set.
Properties

**segmentationTime**

The length in seconds of each segment. Required unless markers is set to None.

- **Type:** number
- **Required:** False
- **Minimum:** 1.0

**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. Can be entered as decimal or hexadecimal values. Each PID specified must be in the range of 32 (or 0x20).8182 (or 0x1ff6).

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

**audioFramesPerPes**

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

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

**fragmentTime**

The length in seconds of each fragment. Only used with EBP markers.

- **Type:** number
- **Required:** False
- **Minimum:** 0.0

**ebpLookaheadMs**

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

**ebpAudioInterval**

When videoAndFixedIntervals is selected, 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. Only available when EBP Cablelabs segmentation markers are selected. Partitions 1 and 2 will always follow the video interval.

Type: M2tsAudioInterval (p. 636)
Required: False

scte35Pid
Packet Identifier (PID) of the SCTE-35 stream in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

Type: string
Required: False

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

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

pmtInterval
The number of milliseconds between instances of this table in the output transport stream. Valid values are 0, 10..1000.

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

pcrPeriod
Maximum time in milliseconds between Program Clock Reference (PCRs) inserted into the transport stream.

Type: integer
Required: False
Minimum: 0
Maximum: 500

segmentationStyle
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 "resetCadence" 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 $segmentationTime seconds. When a segmentation style of "maintainCadence" 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 $segmentationTime seconds. Note that EBP lookahead is a slight exception to this rule.
**Properties**

**Type**: M2tsSegmentationStyle (p. 638)
**Required**: False

**ebif**

If set to passthrough, passes any EBIF data from the input source to this output.

**Type**: M2tsEbifControl (p. 636)
**Required**: False

**audioBufferModel**

When set to dvb, uses DVB buffer model for Dolby Digital audio. When set to atsc, the ATSC model is used.

**Type**: M2tsAudioBufferModel (p. 636)
**Required**: False

**dvbNitSettings**

Inserts DVB Network Information Table (NIT) at the specified table repetition interval.

**Type**: DvbNitSettings (p. 589)
**Required**: False

**absentInputAudioBehavior**

When set to drop, output audio streams will be removed from the program if the selected input audio stream is removed from the input. This allows the output audio configuration to dynamically change based on input configuration. If this is set to encodeSilence, all output audio streams will output encoded silence when not connected to an active input stream.

**Type**: M2tsAbsentInputAudioBehavior (p. 635)
**Required**: False

**timedMetadataBehavior**

When set to passthrough, timed metadata will be passed through from input to output.

**Type**: M2tsTimedMetadataBehavior (p. 645)
**Required**: False

**timedMetadataPid**

Packet Identifier (PID) of the timed metadata stream in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

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

**pmtPid**

Packet Identifier (PID) for the Program Map Table (PMT) in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).
etvSignalPid
Packet Identifier (PID) for input source ETV Signal data to this output. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20).8182 (or 0x1ff6).

bufferModel
If set to multiplex, use multiplex buffer model for accurate interleaving. Setting to bufferModel to none can lead to lower latency, but low-memory devices may not be able to play back the stream without interruptions.

scte35Control
Optionally pass SCTE-35 signals from the input source to this output.

ebpPlacement
Controls placement of EBP on Audio PIDs. If set to videoAndAudioPids, EBP markers will be placed on the video PID and all audio PIDs. If set to videoPid, EBP markers will be placed on only the video PID.

arib
When set to enabled, uses ARIB-compliant field muxing and removes video descriptor.

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.

dvbSdtSettings
Inserts DVB Service Description Table (SDT) at the specified table repetition interval.
**Type:** DvbSdtSettings (p. 589)  
**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. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

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

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

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

**pcrControl**
When set to pcrEveryPesPacket, 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.

**Type:** M2tsPcrControl (p. 637)  
**Required:** False

**videoPid**
Packet Identifier (PID) of the elementary video stream in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

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

**esRateInPes**
Include or exclude the ES Rate field in the PES header.

**Type:** M2tsEsRateInPes (p. 637)  
**Required:** False

**segmentationMarkers**
Inserts segmentation markers at each segmentationTime period. raiSegstart sets the Random Access Indicator bit in the adaptation field. raiAdapt sets the RAI bit and adds the current timecode in the private data bytes. psiSegstart 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. ebpLegacy adds Encoder Boundary Point information to the adaptation field using a legacy proprietary format.

**Type:** M2tsSegmentationMarkers (p. 637)
**Required**: False

**dvbTdtSettings**

Inserts DVB Time and Date Table (TDT) at the specified table repetition interval.

Type: [DvbTdtSettings](p. 594)

**Required**: False

**klv**

If set to passthrough, passes any KLV data from the input source to this output.

Type: [M2tsKlv](p. 637)

**Required**: False

**ccDescriptor**

When set to enabled, generates `captionServiceDescriptor` in PMT.

Type: [M2tsCcDescriptor](p. 636)

**Required**: False

**patInterval**

The number of milliseconds between instances of this table in the output transport stream. Valid values are 0, 10..1000.

Type: integer

Required: False

Minimum: 0

Maximum: 1000

**etvPlatformPid**

Packet Identifier (PID) for input source ETV Platform data to this output. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).

Type: string

Required: False

**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. Can be entered as decimal or hexadecimal values. Each PID specified must be in the range of 32 (or 0x20)..8182 (or 0x1ff6).

Type: string

Required: False

**aribCaptionsPid**

Packet Identifier (PID) for ARIB Captions in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20)..8182 (or 0x1ff6).
scte27Pids

Packet Identifier (PID) for input source SCTE-27 data to this output. Multiple values are accepted, and can be entered in ranges and/or by comma separation. Can be entered as decimal or hexadecimal values. Each PID specified must be in the range of 32 (or 0x20) .. 8182 (or 0x1ff6).

klvDataPids

Packet Identifier (PID) for input source KLV data to this output. Multiple values are accepted, and can be entered in ranges and/or by comma separation. Can be entered as decimal or hexadecimal values. Each PID specified must be in the range of 32 (or 0x20) .. 8182 (or 0x1ff6).

M2tsTimedMetadataBehavior

M2ts Timed Metadata Behavior

- NO_PASSTHROUGH
- PASSTHROUGH

M3u8PcrControl

M3u8 Pcr Control

- CONFIGURED_PCR_PERIOD
- PCR_EVERY_PES_PACKET

M3u8Scte35Behavior

M3u8 Scte35 Behavior

- NO_PASSTHROUGH
- PASSTHROUGH

M3u8Settings

Settings information for the .m3u8 container

pmtPid

Packet Identifier (PID) for the Program Map Table (PMT) in the transport stream. Can be entered as a decimal or hexadecimal value.

- Type: string
- Required: False
ecmPid
This parameter is unused and deprecated.

Type: string
Required: False

scte35Behavior
If set to passthrough, passes any SCTE-35 signals from the input source to this output.

Type: M3u8Scte35Behavior (p. 645)
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. Can be entered as a decimal or hexadecimal value.

Type: string
Required: False

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. Can be entered as decimal or hexadecimal values.

Type: string
Required: False

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

Type: integer
Required: False
Minimum: 0

scte35Pid
Packet Identifier (PID) of the SCTE-35 stream in the transport stream. Can be entered as a decimal or hexadecimal value.

Type: string
Required: False

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

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

pcrControl

When set to pcrEveryPesPacket, 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.

Type: M3u8PcrControl (p. 645)
Required: False

videoPid

Packet Identifier (PID) of the elementary video stream in the transport stream. Can be entered as a decimal or hexadecimal value.

Type: string
Required: False

pcrPeriod

Maximum time in milliseconds between Program Clock References (PCRs) inserted into the transport stream.

Type: integer
Required: False
Minimum: 0
Maximum: 500

pmtInterval

The number of milliseconds between instances of this table in the output transport stream. A value of \"0\" writes out the PMT once per segment file.

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

programNum

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

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

patInterval

The number of milliseconds between instances of this table in the output transport stream. A value of \"0\" writes out the PMT once per segment file.

Type: integer
Required: False
Minimum: 0
Maximum: 1000
timedMetadataPid
Packet Identifier (PID) of the timed metadata stream in the transport stream. Can be entered as a decimal or hexadecimal value. Valid values are 32 (or 0x20).8182 (or 0x1ff6).

  Type: string
  Required: False

timedMetadataBehavior
When set to passthrough, timed metadata is passed through from input to output.

  Type: M3u8TimedMetadataBehavior (p. 648)
  Required: False

M3u8TimedMetadataBehavior
M3u8 Timed Metadata Behavior

  NO_PASSTHROUGH
  PASSTHROUGH

MediaPackageGroupSettings
Media Package Group Settings

destination
MediaPackage channel destination.

  Type: OutputLocationRef (p. 656)
  Required: True

MediaPackageOutputDestinationSettings
Media Package Output Destination Settings

channelId
ID of the channel in MediaPackage that is the destination for this output group. You do not need to specify the individual inputs in MediaPackage; MediaLive will handle the connection of the two MediaLive pipelines to the two MediaPackage inputs. The MediaPackage channel and MediaLive channel must be in the same region.

  Type: string
  Required: False
  MinLength: 1

MediaPackageOutputSettings
Media Package Output Settings

Mp2CodingMode
Mp2 Coding Mode
CODING_MODE_1_0
CODING_MODE_2_0

Mp2Settings

Mp2 Settings

codingMode

The MPEG2 Audio coding mode. Valid values are codingMode10 (for mono) or codingMode20 (for stereo).

Type: Mp2CodingMode (p. 648)
Required: False

bitrate

Average bitrate in bits/second.

Type: number
Required: False

sampleRate

Sample rate in Hz.

Type: number
Required: False

MsSmoothGroupSettings

Ms Smooth Group Settings

fragmentLength

Length of mp4 fragments to generate (in seconds). Fragment length must be compatible with GOP size and framerate.

Type: integer
Required: False
Minimum: 1

eventId

MS Smooth event ID to be sent to the IIS server. Should only be specified if eventIdMode is set to useConfigured.

Type: string
Required: False

timestampOffset

Timestamp offset for the event. Only used if timestampOffsetMode is set to useConfiguredOffset.
**Type**: string
**Required**: False

**segmentationMode**

UseInputSegmentation has been deprecated. The configured segment size is always used.

**Type**: SmoothGroupSegmentationMode (p. 663)
**Required**: False

**numRetries**

Number of retry attempts.

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

**eventStopBehavior**

When set to sendEos, send EOS signal to IIS server when stopping the event

**Type**: SmoothGroupEventStopBehavior (p. 663)
**Required**: False

**acquisitionPointId**

The value of the "Acquisition Point Identity" element used in each message placed in the sparse track. Only enabled if sparseTrackType is not "none".

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

**sparseTrackType**

If set to scte35, use incoming SCTE-35 messages to generate a sparse track in this group of MS-Smooth outputs.

**Type**: SmoothGroupSparseTrackType (p. 663)
**Required**: False

**timestampOffsetMode**

Type of timestamp date offset to use. - useEventStartDate: Use the date the event was started as the offset - useConfiguredOffset: Use an explicitly configured date as the offset

**Type**: SmoothGroupTimestampOffsetMode (p. 663)
**Required**: False

**destination**

Smooth Streaming publish point on an IIS server. Elemental Live acts as a "Push" encoder to IIS.

**Type**: OutputLocationRef (p. 656)
**Required**: True
audioOnlyTimecodeControl
If set to passthrough for an audio-only MS Smooth output, the fragment absolute time will be set to the current timecode. This option does not write timecodes to the audio elementary stream.

Type: SmoothGroupAudioOnlyTimecodeControl (p. 662)
Required: False

collectionRetryInterval
Number of seconds to wait before retrying connection to the IIS server if the connection is lost. Content will be cached during this time and the cache will be delivered to the IIS server once the connection is re-established.

Type: integer
Required: False
Minimum: 0

filecacheDuration
Size in seconds of file cache for streaming outputs.

Type: integer
Required: False
Minimum: 0

certificateMode
If set to verifyAuthenticity, verify the https certificate chain to a trusted Certificate Authority (CA). This will cause https outputs to self-signed certificates to fail.

Type: SmoothGroupCertificateMode (p. 663)
Required: False

inputLossAction
Parameter that control output group behavior on input loss.

Type: InputLossActionForMsSmoothOut (p. 631)
Required: False

sendDelayMs
Number of milliseconds to delay the output from the second pipeline.

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

eventIdMode
Specifies whether or not to send an event ID to the IIS server. If no event ID is sent and the same Live Event is used without changing the publishing point, clients might see cached video from the previous run. Options: - "useConfigured" - use the value provided in eventid - "useTimestamp" - generate and send an event ID based on the current timestamp - "noEventId" - do not send an event ID to the IIS server.
**Properties**

**Type:** SmoothGroupEventIdMode (p. 663)  
**Required:** False

**restartDelay**

Number of seconds before initiating a restart due to output failure, due to exhausting the numRetries on one segment, or exceeding filecacheDuration.

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

**streamManifestBehavior**

When set to send, send stream manifest so publishing point doesn't start until all streams start.

**Type:** SmoothGroupStreamManifestBehavior (p. 663)  
**Required:** False

**MsSmoothOutputSettings**

Ms Smooth Output Settings

**nameModifier**

String concatenated to the end of the destination filename. Required for multiple outputs of the same type.

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

**NetworkInputServerValidation**

Network Input Server Validation

- CHECK_CRYPTOGRAPHY_AND_VALIDATE_NAME  
- CHECK_CRYPTOGRAPHY_ONLY

**NetworkInputSettings**

Network source to transcode. Must be accessible to the Elemental Live node that is running the live event through a network connection.

**hlsInputSettings**

Specifies HLS input settings when the uri is for a HLS manifest.

**Type:** HlsInputSettings (p. 624)  
**Required:** False

**serverValidation**

Check HTTPS server certificates. When set to checkCryptographyOnly, cryptography in the certificate will be checked, but not the server's name. Certain subdomains (notably S3 buckets that use dots in the
bucket name) do not strictly match the corresponding certificate's wildcard pattern and would otherwise cause the event to error. This setting is ignored for protocols that do not use https.

Type: NetworkInputServerValidation (p. 652)
Required: False

Output

Output settings. There can be multiple outputs within a group.

videoDescriptionName

The name of the VideoDescription used as the source for this output.

Type: string
Required: False

outputName

The name used to identify an output.

Type: string
Required: False
MinLength: 1
MaxLength: 255

captionDescriptionNames

The names of the CaptionDescriptions used as caption sources for this output.

Type: Array of type string
Required: False

outputSettings

Output type-specific settings.

Type: OutputSettings (p. 656)
Required: True

audioDescriptionNames

The names of the AudioDescriptions used as audio sources for this output.

Type: Array of type string
Required: False

OutputDestination

mediaPackageSettings

Destination settings for a MediaPackage output; one destination for both encoders.

Type: Array of type MediaPackageOutputDestinationSettings (p. 648)
Required: False

**settings**

Destination settings for a standard output; one destination for each redundant encoder.

- **Type:** Array of type `OutputDestinationSettings` (p. 654)
- **Required:** False

**id**

User-specified id. This is used in an output group or an output.

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

**OutputDestinationSettings**

**passwordParam**

Key used to extract the password from EC2 Parameter store

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

**streamName**

Stream name for RTMP destinations (URLs of type rtmp://)

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

**url**

A URL specifying a destination

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

**username**

Username for destination

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

**OutputGroup**

Output groups for this Live Event. Output groups contain information about where streams should be distributed.

**outputs**

- **Type:** Array of type `Output` (p. 653)
Properties

Required: True

outputGroupSettings

Settings associated with the output group.

Type: OutputGroupSettings (p. 655)
Required: True

name

Custom output group name optionally defined by the user. Only letters, numbers, and the underscore character allowed; only 32 characters allowed.

Type: string
Required: False
MaxLength: 32

OutputGroupSettings

Output Group Settings

archiveGroupSettings

Type: ArchiveGroupSettings (p. 569)
Required: False

mediaPackageGroupSettings

Type: MediaPackageGroupSettings (p. 648)
Required: False

rtmpGroupSettings

Type: RtmpGroupSettings (p. 658)
Required: False

udpGroupSettings

Type: UdpGroupSettings (p. 666)
Required: False

msSmoothGroupSettings

Type: MsSmoothGroupSettings (p. 649)
Required: False

hlsGroupSettings

Type: HlsGroupSettings (p. 618)
Required: False
frameCaptureGroupSettings
  Type: FrameCaptureGroupSettings (p. 603)
  Required: False

OutputLocationRef
Reference to an OutputDestination ID defined in the channel

destinationRefId
  Type: string
  Required: False

OutputSettings
Output Settings

rtmpOutputSettings
  Type: RtmpOutputSettings (p. 659)
  Required: False

archiveOutputSettings
  Type: ArchiveOutputSettings (p. 569)
  Required: False

frameCaptureOutputSettings
  Type: FrameCaptureOutputSettings (p. 603)
  Required: False

msSmoothOutputSettings
  Type: MsSmoothOutputSettings (p. 652)
  Required: False

mediaPackageOutputSettings
  Type: MediaPackageOutputSettings (p. 648)
  Required: False

udpOutputSettings
  Type: UdpOutputSettings (p. 666)
  Required: False

hlsOutputSettings
  Type: HlsOutputSettings (p. 626)
Properties

Required: False

PassThroughSettings
Pass Through Settings

RemixSettings
Remix Settings

channelMappings
Mapping of input channels to output channels, with appropriate gain adjustments.

Type: Array of type AudioChannelMapping (p. 570)
Required: True

channelsOut
Number of output channels to be produced. Valid values: 1, 2, 4, 6, 8

Type: integer
Required: False
Minimum: 1
Maximum: 8

channelsIn
Number of input channels to be used.

Type: integer
Required: False
Minimum: 1
Maximum: 16

ResourceConflict

message

Type: string
Required: False

ResourceNotFound

message

Type: string
Required: False

RtmpCacheFullBehavior

Rtmp Cache Full Behavior
DISCONNECT_IMMEDIATELY
WAIT_FOR_SERVER

**RtmpCaptionData**

Rtmp Caption Data

- ALL
- FIELD1_608
- FIELD1_AND_FIELD2_608

**RtmpCaptionInfoDestinationSettings**

Rtmp Caption Info Destination Settings

**RtmpGroupSettings**

Rtmp Group Settings

**inputLossAction**

Controls the behavior of this RTMP group if input becomes unavailable. - emitOutput: Emit a slate until input returns. - pauseOutput: Stop transmitting data until input returns. This does not close the underlying RTMP connection.

- **Type:** InputLossActionForRtmpOut (p. 631)
- **Required:** False

**captionData**

Controls the types of data that passes to onCaptionInfo outputs. If set to 'all' then 608 and 708 carried DTVCC data will be passed. If set to 'field1AndField2608' then DTVCC data will be stripped out, but 608 data from both fields will be passed. If set to 'field1608' then only the data carried in 608 from field 1 video will be passed.

- **Type:** RtmpCaptionData (p. 658)
- **Required:** False

**authenticationScheme**

Authentication scheme to use when connecting with CDN

- **Type:** AuthenticationScheme (p. 576)
- **Required:** False

**cacheLength**

Cache length, in seconds, is used to calculate buffer size.

- **Type:** integer
- **Required:** False
- **Minimum:** 30
restartDelay
If a streaming output fails, number of seconds to wait until a restart is initiated. A value of 0 means never restart.

Type: integer
Required: False
Minimum: 0

cacheFullBehavior
Controls behavior when content cache fills up. If remote origin server stalls the RTMP connection and does not accept content fast enough the 'Media Cache' will fill up. When the cache reaches the duration specified by cacheLength the cache will stop accepting new content. If set to disconnectImmediately, the RTMP output will force a disconnect. Clear the media cache, and reconnect after restartDelay seconds. If set to waitForServer, the RTMP output will wait up to 5 minutes to allow the origin server to begin accepting data again.

Type: RtmpCacheFullBehavior (p. 657)
Required: False

RtmpOutputCertificateMode
Rtmp Output Certificate Mode

SELF_SIGNED
VERIFY_AUTHENTICITY

RtmpOutputSettings
Rtmp Output Settings
certificateMode
If set to verifyAuthenticity, verify the tls certificate chain to a trusted Certificate Authority (CA). This will cause rtmps outputs with self-signed certificates to fail.

Type: RtmpOutputCertificateMode (p. 659)
Required: False

numRetries
Number of retry attempts.

Type: integer
Required: False
Minimum: 0

destination
The RTMP endpoint excluding the stream name (eg. rtmp://host/appname). For connection to Akamai, a username and password must be supplied. URI fields accept format identifiers.

Type: OutputLocationRef (p. 656)
**Required**: True

**connectionRetryInterval**
Number of seconds to wait before retrying a connection to the Flash Media server if the connection is lost.

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

**Scte20Convert608To708**
Scte20 Convert608 To708

- **DISABLED**
- **UPCONVERT**

**Scte20PlusEmbeddedDestinationSettings**
Scte20 Plus Embedded Destination Settings

**Scte20SourceSettings**
Scte20 Source Settings

**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**
If 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.

- **Type**: Scte20Convert608To708 (p. 660)
- **Required**: False

**Scte27DestinationSettings**
Scte27 Destination Settings

**Scte27SourceSettings**
Scte27 Source Settings
**pid**

The pid field is used in conjunction with the caption selector languageCode field as follows:
- Specify PID and Language: Extracts captions from that PID; the language is "informational".
- Specify PID and omit Language: Extracts the specified PID.
- Omit PID and specify Language: Extracts the specified language, whichever PID that happens to be.
- Omit PID and omit Language: Valid only if source is DVB-Sub that is being passed through; all languages will be passed through.

Type: integer
Required: False
Minimum: 1

**Scte35AposNoRegionalBlackoutBehavior**

Scte35 Apos No Regional Blackout Behavior

FOLLOW
IGNORE

**Scte35AposWebDeliveryAllowedBehavior**

Scte35 Apos Web Delivery Allowed Behavior

FOLLOW
IGNORE

**Scte35SpliceInsert**

Scte35 Splice Insert

**adAvailOffset**

When specified, this offset (in milliseconds) is added to the input Ad Avail PTS time. This only applies to embedded SCTE 104/35 messages and does not apply to OOB messages.

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

**webDeliveryAllowedFlag**

When set to ignore, Segment Descriptors with webDeliveryAllowedFlag set to 0 will no longer trigger blackouts or Ad Avail slates

Type: Scte35SpliceInsertWebDeliveryAllowedBehavior (p. 662)
Required: False

**noRegionalBlackoutFlag**

When set to ignore, Segment Descriptors with noRegionalBlackoutFlag set to 0 will no longer trigger blackouts or Ad Avail slates
Properties

**Type:** Scte35SpliceInsertNoRegionalBlackoutBehavior (p. 662)
**Required:** False

**Scte35SpliceInsertNoRegionalBlackoutBehavior**

Scte35 Splice Insert No Regional Blackout Behavior

- FOLLOW
- IGNORE

**Scte35SpliceInsertWebDeliveryAllowedBehavior**

Scte35 Splice Insert Web Delivery Allowed Behavior

- FOLLOW
- IGNORE

**Scte35TimeSignalApos**

Scte35 Time Signal Apos

**adAvailOffset**

When specified, this offset (in milliseconds) is added to the input Ad Avail PTS time. This only applies to embedded SCTE 104/35 messages and does not apply to OOB messages.

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

**webDeliveryAllowedFlag**

When set to ignore, Segment Descriptors with webDeliveryAllowedFlag set to 0 will no longer trigger blackouts or Ad Avail slates

- **Type:** Scte35AposWebDeliveryAllowedBehavior (p. 661)
- **Required:** False

**noRegionalBlackoutFlag**

When set to ignore, Segment Descriptors with noRegionalBlackoutFlag set to 0 will no longer trigger blackouts or Ad Avail slates

- **Type:** Scte35AposNoRegionalBlackoutBehavior (p. 661)
- **Required:** False

**SmoothGroupAudioOnlyTimecodeControl**

Smooth Group Audio Only Timecode Control

- PASSTHROUGH
USE_CONFIGURED_CLOCK

SmoothGroupCertificateMode
Smooth Group Certificate Mode

SELF_SIGNED
VERIFY_AUTHENTICITY

SmoothGroupEventIdMode
Smooth Group Event Id Mode

NO_EVENT_ID
USE_CONFIGURED
USE_TIMESTAMP

SmoothGroupEventStopBehavior
Smooth Group Event Stop Behavior

NONE
SEND_EOS

SmoothGroupSegmentationMode
Smooth Group Segmentation Mode

USE_INPUT_SEGMENTATION
USE_SEGMENT_DURATION

SmoothGroupSparseTrackType
Smooth Group Sparse Track Type

NONE
SCTE_35

SmoothGroupStreamManifestBehavior
Smooth Group Stream Manifest Behavior

DO_NOT_SEND
SEND

SmoothGroupTimestampOffsetMode
Smooth Group Timestamp Offset Mode

USE_CONFIGURED_OFFSET
USE_EVENT_START_DATE
SmpteTtDestinationSettings

Smpte Tt Destination Settings

StandardHlsSettings

Standard Hls Settings

m3u8Settings

Type: M3u8Settings (p. 645)
Required: True

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

StaticKeySettings

Static Key Settings

staticKeyValue

Static key value as a 32 character hexadecimal string.

Type: string
Required: True
MinLength: 32
MaxLength: 32

keyProviderServer

The URL of the license server used for protecting content.

Type: InputLocation (p. 630)
Required: False

Tags

key-value pairs

Type: string

TeletextDestinationSettings

Teletext Destination Settings

TeletextSourceSettings

Teletext Source Settings
pageNumber

Specifies the teletext page number within the data stream from which to extract captions. Range of 0x100 (256) to 0x8FF (2303). Unused for passthrough. Should be specified as a hexadecimal string with no "0x" prefix.

Type: string
Required: False

TimecodeConfig

Timecode Config

syncThreshold

Threshold in frames beyond which output timecode is resynchronized to the input timecode. Discrepancies below this threshold are permitted to avoid unnecessary discontinuities in the output timecode. No timecode sync when this is not specified.

Type: integer
Required: False
Minimum: 1
Maximum: 1000000

source

Identifies the source for the timecode that will be associated with the events outputs. -Embedded (embedded): Initialize the output timecode with timecode from the the source. If no embedded timecode is detected in the source, the system falls back to using "Start at 0" (zerobased). -System Clock (systemclock): Use the UTC time. -Start at 0 (zerobased): The time of the first frame of the event will be 00:00:00:00.

Type: TimecodeConfigSource (p. 665)
Required: True

TimecodeConfigSource

Timecode Config Source

EMBEDDED
SYSTEMCLOCK
ZEROBASED

TtmlDestinationSettings

Ttml Destination Settings

styleControl

When set to passthrough, passes through style and position information from a TTML-like input source (TTML, SMPTE-TT, CFF-TT) to the CFF-TT output or TTML output.

Type: TtmlDestinationStyleControl (p. 666)
Required: False
**TtmlDestinationStyleControl**

Ttml Destination Style Control

- PASSTHROUGH
- USE_CONFIGURED

**UdpContainerSettings**

Udp Container Settings

**m2tsSettings**

- **Type**: M2tsSettings (p. 638)
- **Required**: False

**UdpGroupSettings**

Udp Group Settings

**inputLossAction**

Specifies behavior of last resort when input video is lost, and no more backup inputs are available. When dropTs is selected the entire transport stream will stop being emitted. When dropProgram is selected the program can be dropped from the transport stream (and replaced with null packets to meet the TS bitrate requirement). Or, when emitProgram is chosen the transport stream will continue to be produced normally with repeat frames, black frames, or slate frames substituted for the absent input video.

- **Type**: InputLossActionForUdpOut (p. 631)
- **Required**: False

**timedMetadataId3Frame**

Indicates ID3 frame that has the timecode.

- **Type**: UdpTimedMetadataId3Frame (p. 667)
- **Required**: False

**timedMetadataId3Period**

Timed Metadata interval in seconds.

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

**UdpOutputSettings**

Udp Output Settings

**destination**

Destination address and port number for RTP or UDP packets. Can be unicast or multicast RTP or UDP (eg. rtp://239.10.10.10:5001 or udp://10.100.100.100:5002).
**Properties**

**bufferMsec**

UDP output buffering in milliseconds. Larger values increase latency through the transcoder but simultaneously assist the transcoder in maintaining a constant, low-jitter UDP/RTP output while accommodating clock recovery, input switching, input disruptions, picture reordering, etc.

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

**containerSettings**

- **Type**: UdpContainerSettings (p. 666)
- **Required**: True

**fecOutputSettings**

Settings for enabling and adjusting Forward Error Correction on UDP outputs.

- **Type**: FecOutputSettings (p. 602)
- **Required**: False

**UdpTimedMetadataId3Frame**

Udp Timed Metadata Id3 Frame

- NONE
- PRIV
- TDRL

**VideoCodecSettings**

Video Codec Settings

**h264Settings**

- **Type**: H264Settings (p. 608)
- **Required**: False

**frameCaptureSettings**

- **Type**: FrameCaptureSettings (p. 604)
- **Required**: False

**VideoDescription**

Video settings for this stream.
respondToAfd

Indicates how to respond to the AFD values in the input stream. RESPOND causes input video to be clipped, depending on the AFD value, input display aspect ratio, and output display aspect ratio, and (except for FRAMECAPTURE codec) includes the values in the output. PASSTHROUGH (does not apply to FRAMECAPTURE codec) ignores the AFD values and includes the values in the output, so input video is not clipped. NONE ignores the AFD values and does not include the values through to the output, so input video is not clipped.

- **Type**: VideoDescriptionRespondToAfd (p. 669)
- **Required**: False

scalingBehavior

STRETCHTOOUTPUT configures the output position to stretch the video to the specified output resolution (height and width). This option will override any position value. DEFAULT may insert black boxes (pillar boxes or letter boxes) around the video to provide the specified output resolution.

- **Type**: VideoDescriptionScalingBehavior (p. 669)
- **Required**: False

name

The name of this VideoDescription. Outputs will use this name to uniquely identify this Description. Description names should be unique within this Live Event.

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

width

Output video width, in pixels. Must be an even number. For most codecs, you can leave this field and height blank in order to use the height and width (resolution) from the source. Note, however, that leaving blank is not recommended. For the Frame Capture codec, height and width are required.

- **Type**: integer
- **Required**: False

sharpness

Changes the strength of the anti-alias filter used for scaling. 0 is the softest setting, 100 is the sharpest. A setting of 50 is recommended for most content.

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

codecSettings

Video codec settings.

- **Type**: VideoCodecSettings (p. 667)
- **Required**: False
**height**

Output video height, in pixels. Must be an even number. For most codecs, you can leave this field and width blank in order to use the height and width (resolution) from the source. Note, however, that leaving blank is not recommended. For the Frame Capture codec, height and width are required.

- **Type:** integer
- **Required:** False

**VideoDescriptionRespondToAfd**

Video Description Respond To Afd

- NONE
- PASSTHROUGH
- RESPOND

**VideoDescriptionScalingBehavior**

Video Description Scaling Behavior

- DEFAULT
- STRETCH_TO_OUTPUT

**VideoSelector**

Specifies a particular video stream within an input source. An input may have only a single video selector.

**colorSpace**

Specifies the colorspace of an input. This setting works in tandem with colorSpaceConversion to determine if any conversion will be performed.

- **Type:** VideoSelectorColorSpace (p. 670)
- **Required:** False

**selectorSettings**

The video selector settings.

- **Type:** VideoSelectorSettings (p. 670)
- **Required:** False

**colorSpaceUsage**

Applies only if colorSpace is a value other than follow. This field controls how the value in the colorSpace field will be used. fallback means that when the input does include color space data, that data will be used, but when the input has no color space data, the value in colorSpace will be used. Choose fallback if your input is sometimes missing color space data, but when it does have color space data, that data is correct. force means to always use the value in colorSpace. Choose force if your input usually has no color space data or might have unreliable color space data.

- **Type:** VideoSelectorColorSpaceUsage (p. 670)
- **Required:** False
**VideoSelectorColorSpace**

Video Selector Color Space

- FOLLOW
- REC_601
- REC_709

**VideoSelectorColorSpaceUsage**

Video Selector Color Space Usage

- FALLBACK
- FORCE

**VideoSelectorPid**

Video Selector Pid

**pid**

Selects a specific PID from within a video source.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
- **Maximum:** 8191

**VideoSelectorProgramId**

Video Selector Program Id

**programId**

Selects a specific program from within a multi-program transport stream. If the program doesn't exist, the first program within the transport stream will be selected by default.

- **Type:** integer
- **Required:** False
- **Minimum:** 0
- **Maximum:** 65536

**VideoSelectorSettings**

Video Selector Settings

**videoSelectorPid**

- **Type:** VideoSelectorPid (p. 670)
- **Required:** False

**videoSelectorProgramId**

- **Type:** VideoSelectorProgramId (p. 670)
**InputSecurityGroups**

**URI**
/prod/inputSecurityGroups

**HTTP Methods**

GET

Operation ID: ListInputSecurityGroups

Produces a list of Input Security Groups for an account

**Query Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>

**Responses**

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>ListInputSecurityGroupsResultModel (p. 672)</td>
<td>An array of Input Security Groups</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 673)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 673)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 673)</td>
<td>Request limit exceeded on list channel calls to channel service.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 673)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>BadGatewayException (p. 674)</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>504</td>
<td>GatewayTimeoutException (p. 674)</td>
<td>Gateway Timeout Error</td>
</tr>
</tbody>
</table>

**POST**

Operation ID: CreateInputSecurityGroup
Creates a Input Security Group

Responses

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>InvalidRequest (p. 673)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 673)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 673)</td>
<td>Request limit exceeded on list channel calls to channel service.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 673)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>BadGatewayException (p. 673)</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>504</td>
<td>GatewayTimeoutException (p. 674)</td>
<td>Gateway Timeout Error</td>
</tr>
</tbody>
</table>

Schemas

Request Bodies

Example POST

```json
{
    "whitelistRules": [
        {
            "cidr": "string"
        }
    ],
    "tags": {
    }
}
```

Response Bodies

Example ListInputSecurityGroupsResultModel

```json
{
    "inputSecurityGroups": [
        {
            "inputs": [
                "string"
            ],
            "id": "string",
            "state": enum,
            "arn": "string",
            "whitelistRules": [
                {
                    "cidr": "string"
                }
            ],
            "inputs": ["string"]
        }
    ],
```
"tags": {
  
},
"nextToken": "string"
}

**Example CreateInputSecurityGroupResultModel**

```json
{
  "securityGroup": {
    "inputs": [
      "string"
    ],
    "id": "string",
    "state": enum,
    "arn": "string",
    "whitelistRules": [
      {
        "cidr": "string"
      }
    ],
    "tags": {
      
    }
  }
}
```

**Example InvalidRequest**

```json
{
  "message": "string"
}
```

**Example AccessDenied**

```json
{
  "message": "string"
}
```

**Example LimitExceeded**

```json
{
  "message": "string"
}
```

**Example InternalServiceError**

```json
{
  "message": "string"
}
```

**Example BadGatewayException**

```json
{
  "message": "string"
}
```
Example GatewayTimeoutException

```json
{
  "message": "string"
}
```

Properties

AccessDenied

message

Type: string
Required: False

BadGatewayException

message

Type: string
Required: False

CreateInputSecurityGroupResultModel

securityGroup

Type: InputSecurityGroup (p. 674)
Required: False

GatewayTimeoutException

message

Type: string
Required: False

InputSecurityGroup

An Input Security Group

inputs

The list of inputs currently using this Input Security Group.

Type: Array of type string
Required: False

id

The Id of the Input Security Group

Type: string
Properties

**state**
The current state of the Input Security Group.

*Type: InputSecurityGroupState (p. 675)*
*Required: False*

**arn**
Unique ARN of Input Security Group

*Type: string*
*Required: False*

**whitelistRules**
Whitelist rules and their sync status

*Type: Array of type InputWhitelistRule (p. 676)*
*Required: False*

**tags**
A collection of key-value pairs.

*Type: Tags (p. 677)*
*Required: False*

---

**InputSecurityGroupState**

IDLE
IN_USE
UPDATING
DELETED

**InputSecurityGroupWhitelistRequest**

Request of IPv4 CIDR addresses to whitelist in a security group.

**whitelistRules**
List of IPv4 CIDR addresses to whitelist

*Type: Array of type InputWhitelistRuleCidr (p. 676)*
*Required: False*

**tags**
A collection of key-value pairs.

*Type: Tags (p. 677)*
*Required: False*
**InputWhitelistRule**

Whitelist rule

**cidr**

The IPv4 CIDR that's whitelisted.

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

**InputWhitelistRuleCidr**

An IPv4 CIDR to whitelist.

**cidr**

The IPv4 CIDR to whitelist.

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

**InternalServerError**

**message**

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

**InvalidRequest**

**message**

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

**LimitExceeded**

**message**

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

**ListInputSecurityGroupsResultModel**

Result of input security group list request

**inputSecurityGroups**

List of input security groups

  * **Type:** Array of type InputSecurityGroup (p. 674)
  * **Required:** False
nextToken

Type: string
Required: False

Tags
key-value pairs
Type: string

InputSecurityGroups inputSecurityGroupId

URI
/prod/inputSecurityGroups/{inputSecurityGroupId}

HTTP Methods
GET

Operation ID: DescribeInputSecurityGroup

Produces a summary of an Input Security Group

Path Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>inputSecurityGroupId</td>
<td>String</td>
<td>True</td>
<td>The id of the Input Security Group to describe</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>InputSecurityGroup (p. 679)</td>
<td>An Input Security Group</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 680)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 680)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>404</td>
<td>ResourceNot Found (p. 680)</td>
<td>The channel you're requesting to describe does not exist.</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 680)</td>
<td>Request limit exceeded on list channel calls to channel service.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 681)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>BadGatewayException (p. 681)</td>
<td>Bad Gateway Error</td>
</tr>
</tbody>
</table>
### PUT

Operation ID: UpdateInputSecurityGroup


#### Path Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>inputSecurityGroupId</td>
<td>String</td>
<td>True</td>
<td>The id of the Input Security Group to describe</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>InvalidRequest (p. 680)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 680)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>404</td>
<td>ResourceNotFoundException (p. 680)</td>
<td>The channel you're requesting to describe does not exist.</td>
</tr>
<tr>
<td>409</td>
<td>ResourceConflict (p. 680)</td>
<td>The channel is unable to create due to an issue with channel resources.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 680)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>BadGatewayException (p. 680)</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>504</td>
<td>GatewayTimeoutException (p. 680)</td>
<td>Gateway Timeout Error</td>
</tr>
</tbody>
</table>

### DELETE

Operation ID: DeleteInputSecurityGroup

Deletes an Input Security Group

#### Path Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>inputSecurityGroupId</td>
<td>String</td>
<td>True</td>
<td>The id of the Input Security Group to describe</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>Empty (p. 680)</td>
<td>An Input Security Group</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 680)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 680)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>404</td>
<td>ResourceNotFound (p. 680)</td>
<td>The channel you’re requesting to describe does not exist.</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 680)</td>
<td>Request limit exceeded on list channel calls to channel service.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 681)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>BadGatewayException (p. 681)</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>504</td>
<td>GatewayTimeoutException (p. 681)</td>
<td>Gateway Timeout Error</td>
</tr>
</tbody>
</table>

**Schemas**

**Request Bodies**

**Example PUT**

```json
{
    "whitelistRules": [
        {
            "cidr": "string"
        }
    ],
    "tags": {}
}
```

**Response Bodies**

**Example InputSecurityGroup**

```json
{
    "inputs": [
        "String"
    ],
    "id": "string",
    "state": enum,
    "arn": "string",
    "whitelistRules": [
        {
            "cidr": "string"
        }
    ],
    "tags": {}
}
```
Example UpdateInputSecurityGroupResultModel

```json
{
  "securityGroup": {
    "inputs": [
      "string"
    ],
    "id": "string",
    "state": enum,
    "arn": "string",
    "whitelistRules": [
      {
        "cidr": "string"
      }
    ],
    "tags": {
    }
  }
}
```

Example Empty

```json
{
}
```

Example InvalidRequest

```json
{
  "message": "string"
}
```

Example AccessDenied

```json
{
  "message": "string"
}
```

Example ResourceNotFound

```json
{
  "message": "string"
}
```

Example ResourceConflict

```json
{
  "message": "string"
}
```

Example LimitExceeded

```json
{
  "message": "string"
}
```
Example InternalServiceError

```json
{
  "message": "string"
}
```

Example BadGatewayException

```json
{
  "message": "string"
}
```

Example GatewayTimeoutException

```json
{
  "message": "string"
}
```

Properties

AccessDenied

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

BadGatewayException

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

Empty

GatewayTimeoutException

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

InputSecurityGroup

An Input Security Group

- **inputs**
  - The list of inputs currently using this Input Security Group.
Properties

**Type**
Array of type string

**Required**
False

**id**
The Id of the Input Security Group

**Type**
string

**Required**
False

**state**
The current state of the Input Security Group.

**Type**
InputSecurityGroupState (p. 682)

**Required**
False

**arn**
Unique ARN of Input Security Group

**Type**
string

**Required**
False

**whitelistRules**
Whitelist rules and their sync status

**Type**
Array of type InputWhitelistRule (p. 683)

**Required**
False

**tags**
A collection of key-value pairs.

**Type**
Tags (p. 684)

**Required**
False

**InputSecurityGroupState**

- IDLE
- IN_USE
- UPDATING
- DELETED

**InputSecurityGroupWhitelistRequest**
Request of IPv4 CIDR addresses to whitelist in a security group.

**whitelistRules**
List of IPv4 CIDR addresses to whitelist
**Properties**

- **Type**: Array of type `InputWhitelistRuleCidr (p. 683)`  
  **Required**: False

**tags**

A collection of key-value pairs.

- **Type**: Tags (p. 684)  
  **Required**: False

**InputWhitelistRule**

Whitelist rule

- **cidr**
  
  The IPv4 CIDR that's whitelisted.

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

**InputWhitelistRuleCidr**

An IPv4 CIDR to whitelist.

- **cidr**
  
  The IPv4 CIDR to whitelist.

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

**InternalServiceError**

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

**InvalidRequest**

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

**LimitExceeded**

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

**ResourceConflict**

message

Type: string
Required: False

**ResourceNotFound**

message

Type: string
Required: False

**Tags**

key-value pairs

Type: string

**UpdateInputSecurityGroupResultModel**

securityGroup

Type: InputSecurityGroup (p. 681)
Required: False

**Inputs**

**URI**

/prod/inputs

**HTTP Methods**

**GET**

Operation ID: ListInputs

Produces list of inputs that have been created

**Query Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>
**Responses**

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>ListInputsResultModel (p. 686)</td>
<td>An array of inputs</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 688)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 688)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 688)</td>
<td>Request limit exceeded on list channel calls to channel service.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 688)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>BadGatewayException (p. 688)</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>504</td>
<td>GatewayTimeoutException (p. 688)</td>
<td>Gateway Timeout Error</td>
</tr>
</tbody>
</table>

**POST**

Operation ID: CreateInput

Create an input

**Responses**

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>CreateInputResultModel (p. 687)</td>
<td>Creation of channel is started.</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 688)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 688)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 688)</td>
<td>Request limit exceeded on list channel calls to channel service.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 688)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>BadGatewayException (p. 688)</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>504</td>
<td>GatewayTimeoutException (p. 688)</td>
<td>Gateway Timeout Error</td>
</tr>
</tbody>
</table>

**Schemas**

**Request Bodies**

**Example POST**

```json
{
  "inputSecurityGroups": [
    "string"
  ]
}```
Response Bodies

Example ListInputsResultModel

```json
{
    "inputs": [
        {
            "sources": [
                {
                    "passwordParam": "string",
                    "url": "string",
                    "username": "string"
                }
            ],
            "destinations": [
                {
                    "port": "string",
                    "ip": "string",
                    "vpc": {
                        "networkInterfaceId": "string",
                        "availabilityZone": "string"
                    },
                    "url": "string"
                }
            ],
            "type": enum,
            "tags": {
            },
            "mediaConnectFlows": [
```
Example CreateInputResultModel

```json
{
  "input": {
    "sources": [
      {
        "passwordParam": "string",
        "url": "string",
        "username": "string"
      }
    ],
    "destinations": [
      {
        "port": "string",
        "ip": "string",
        "vpc": {
          "networkInterfaceId": "string",
          "availabilityZone": "string"
        },
        "url": "string"
      }
    ],
    "type": enum,
    "tags": {
    },
    "mediaConnectFlows": [
      {
        "flowArn": "string"
      }
    ],
    "attachedChannels": [
      "string"
    ],
    "roleArn": "string",
    "inputClass": enum,
    "name": "string",
    "securityGroups": [
      "string"
    ],
    "id": "string",
    "state": enum,
    "arn": "string"
  }
}
```
Example InvalidRequest

```json
{
  "message": "string"
}
```

Example AccessDenied

```json
{
  "message": "string"
}
```

Example LimitExceeded

```json
{
  "message": "string"
}
```

Example InternalServiceError

```json
{
  "message": "string"
}
```

Example BadGatewayException

```json
{
  "message": "string"
}
```

Example GatewayTimeoutException

```json
{
  "message": "string"
}
```

Properties

AccessDenied

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

BadGatewayException

- **message**
  - Type: string
CreateInput

inputSecurityGroups
A list of security groups referenced by IDs to attach to the input.

  Type: Array of type string
  Required: False

mediaConnectFlows
A list of the MediaConnect Flows that you want to use in this input. You can specify as few as one Flow and presently, as many as two. The only requirement is when you have more than one is that each Flow is in a separate Availability Zone as this ensures your EML input is redundant to AZ issues.

  Type: Array of type MediaConnectFlowRequest (p. 696)
  Required: False

sources
The source URLs for a PULL-type input. Every PULL type input needs exactly two source URLs for redundancy. Only specify sources for PULL type Inputs. Leave Destinations empty.

  Type: Array of type InputSourceRequest (p. 694)
  Required: False

requestId
Unique identifier of the request to ensure the request is handled exactly once in case of retries.

  Type: string
  Required: False

roleArn
The Amazon Resource Name (ARN) of the role this input assumes during and after creation.

  Type: string
  Required: False

destinations
Destination settings for PUSH type inputs.

  Type: Array of type InputDestinationRequest (p. 693)
  Required: False

name
Name of the input.

  Type: string
Properties

Required: False

vpc

Type: InputVpcRequest (p. 695)
Required: False

type

Type: InputType (p. 694)
Required: False

tags

A collection of key-value pairs.

Type: Tags (p. 696)
Required: False

CreateInputResultModel

input

Type: Input (p. 690)
Required: False

GatewayTimeoutException

message

Type: string
Required: False

Input

sources

A list of the sources of the input (PULL-type).

Type: Array of type InputSource (p. 693)
Required: False

destinations

A list of the destinations of the input (PUSH-type).

Type: Array of type InputDestination (p. 692)
Required: False

type

Type: InputType (p. 694)
**Required**: False

**tags**
A collection of key-value pairs.

**Type**: Tags (p. 696)
**Required**: False

**mediaConnectFlows**
A list of MediaConnect Flows for this input.

**Type**: Array of type MediaConnectFlow (p. 696)
**Required**: False

**attachedChannels**
A list of channel IDs that that input is attached to (currently an input can only be attached to one channel).

**Type**: Array of type string
**Required**: False

**roleArn**
The Amazon Resource Name (ARN) of the role this input assumes during and after creation.

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

**inputClass**
STANDARD - MediaLive expects two sources to be connected to this input. If the channel is also STANDARD, both sources will be ingested. If the channel is SINGLE_PIPELINE, only the first source will be ingested; the second source will always be ignored, even if the first source fails. SINGLE_PIPELINE - You can connect only one source to this input. If the ChannelClass is also SINGLE_PIPELINE, this value is valid. If the ChannelClass is STANDARD, this value is not valid because the channel requires two sources in the input.

**Type**: InputClass (p. 692)
**Required**: False

**name**
The user-assigned name (This is a mutable value).

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

**securityGroups**
A list of IDs for all the Input Security Groups attached to the input.
Properties

Type: Array of type string
Required: False

id

The generated ID of the input (unique for user account, immutable).

Type: string
Required: False

state

Type: InputState (p. 694)
Required: False

arn

The Unique ARN of the input (generated, immutable).

Type: string
Required: False

InputClass

A standard input has two sources and a single pipeline input only has one.

STANDARD
SINGLE_PIPELINE

InputDestination

The settings for a PUSH type input.

port

The port number for the input.

Type: string
Required: False

ip

The system-generated static IP address of endpoint. It remains fixed for the lifetime of the input.

Type: string
Required: False

vpc

Type: InputDestinationVpc (p. 693)
Required: False
url

This represents the endpoint that the customer stream will be pushed to.

    Type: string
    Required: False

**InputDestinationRequest**

Endpoint settings for a PUSH type input.

**streamName**

A unique name for the location the RTMP stream is being pushed to.

    Type: string
    Required: False

**InputDestinationVpc**

The properties for a VPC type input destination.

**networkInterfaceId**

The network interface ID of the Input destination in the VPC.

    Type: string
    Required: False

**availabilityZone**

The availability zone of the Input destination.

    Type: string
    Required: False

**InputSource**

The settings for a PULL type input.

**passwordParam**

The key used to extract the password from EC2 Parameter store.

    Type: string
    Required: False

url

This represents the customer's source URL where stream is pulled from.

    Type: string
Required: False

**username**

The username for the input source.

Type: string
Required: False

**InputSourceRequest**

Settings for a PULL type input.

**passwordParam**

The key used to extract the password from EC2 Parameter store.

Type: string
Required: False

**url**

This represents the customer's source URL where stream is pulled from.

Type: string
Required: False

**username**

The username for the input source.

Type: string
Required: False

**InputState**

CREATING
DETACHED
ATTACHED
DELETING
DELETED

**InputType**

UDP_PUSH
RTP_PUSH
RTMP_PUSH
RTMP_PULL
URL_PULL
MP4_FILE
MEDIACONNECT
### InputVpcRequest

Settings for a private VPC Input. When this property is specified, the input destination addresses will be created in a VPC rather than with public Internet addresses. This property requires setting the roleArn property on Input creation. Not compatible with the inputSecurityGroups property.

**securityGroupIds**

A list of up to 5 EC2 VPC security group IDs to attach to the Input VPC network interfaces. Requires subnetIds. If none are specified then the VPC default security group will be used.

- **Type:** Array of type string
- **Required:** False

**subnetIds**

A list of 2 VPC subnet IDs from the same VPC. Subnet IDs must be mapped to two unique availability zones (AZ).

- **Type:** Array of type string
- **Required:** True

### InternalServiceError

**message**

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

### InvalidRequest

**message**

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

### LimitExceeded

**message**

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

### ListInputsResultModel

**inputs**

- **Type:** Array of type Input (p. 690)
- **Required:** False

**nextToken**

- **Type:** string
**Inputs inputId**

**URI**
/prod/inputs/inputId

**HTTP Methods**

**GET**

Operation ID: DescribeInput

Produces details about an input

**Path Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>inputId</td>
<td>String</td>
<td>True</td>
<td>Unique ID of the input</td>
</tr>
</tbody>
</table>
PUT

Operation ID: UpdateInput

Updates an input.

Path Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>inputId</td>
<td>String</td>
<td>True</td>
<td>Unique ID of the input</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>UpdateInputResultModel (p. 699)</td>
<td>The input update is successfully initiated.</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 700)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 700)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>404</td>
<td>ResourceNotFound (p. 701)</td>
<td>The channel you're requesting to describe does not exist.</td>
</tr>
<tr>
<td>409</td>
<td>ResourceConflict (p. 701)</td>
<td>The channel is unable to create due to an issue with channel resources.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 701)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>BadGatewayException (p. 701)</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>504</td>
<td>GatewayTimeoutException (p. 701)</td>
<td>Gateway Timeout Error</td>
</tr>
</tbody>
</table>
DELETE

Operation ID: DeleteInput

Deletes the input end point

Path Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>inputId</td>
<td>String</td>
<td>True</td>
<td>Unique ID of the input</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>Empty (p. 700)</td>
<td>Successful deletion</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 700)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 700)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>404</td>
<td>ResourceNotFound (p. 701)</td>
<td>The channel you’re requesting to describe does not exist.</td>
</tr>
<tr>
<td>409</td>
<td>ResourceConflict (p. 701)</td>
<td>The channel is unable to create due to an issue with channel resources.</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 701)</td>
<td>Request limit exceeded on list channel calls to channel service.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 701)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>BadGatewayException (p. 701)</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>504</td>
<td>GatewayTimeoutException (p. 701)</td>
<td>Gateway Timeout Error</td>
</tr>
</tbody>
</table>

Schemas

Request Bodies

Example PUT

```json
{
    "inputSecurityGroups": [
        "string"
    ],
    "mediaConnectFlows": [
        
```
"flowArn": "string"
},
"sources": [
{
"passwordParam": "string",
"url": "string",
"username": "string"
}
],
"roleArn": "string",
"destinations": [
{
"streamName": "string"
}
],
"name": "string"
}

Response Bodies

Example Input

{
"sources": [
{
"passwordParam": "string",
"url": "string",
"username": "string"
}
],
"destinations": [
{
"port": "string",
"ip": "string",
"vpc": {
"networkInterfaceId": "string",
"availabilityZone": "string"
},
"url": "string"
}
],
"type": enum,
"tags": {
},
"mediaConnectFlows": [
{
"flowArn": "string"
}
],
"attachedChannels": [
"string"
],
"roleArn": "string",
"inputClass": enum,
"name": "string",
"securityGroups": [
"string"
],
"id": "string",
"state": enum,
"arn": "string"}
Example UpdateInputResultModel

```
{
  "input": {
    "sources": [
      {
        "passwordParam": "string",
        "url": "string",
        "username": "string"
      }
    ],
    "destinations": [
      {
        "port": "string",
        "ip": "string",
        "vpc": {
          "networkInterfaceId": "string",
          "availabilityZone": "string"
        },
        "url": "string"
      }
    ],
    "type": enum,
    "tags": {
    },
    "mediaConnectFlows": [
      {
        "flowArn": "string"
      }
    ],
    "attachedChannels": ["string"
]
  },
  "roleArn": "string",
  "inputClass": enum,
  "name": "string",
  "securityGroups": ["string"
]
  },
  "id": "string",
  "state": enum,
  "arn": "string"
}
```

Example Empty

```
{
}
```

Example InvalidRequest

```
{
  "message": "string"
}
```

Example AccessDenied

```
{
  "message": "string"
}
```
Example ResourceNotFound

```json
{
  "message": "string"
}
```

Example ResourceConflict

```json
{
  "message": "string"
}
```

Example LimitExceeded

```json
{
  "message": "string"
}
```

Example InternalServiceError

```json
{
  "message": "string"
}
```

Example BadGatewayException

```json
{
  "message": "string"
}
```

Example GatewayTimeoutException

```json
{
  "message": "string"
}
```

Properties

**AccessDenied**

*message*

*Type:* string
*Required:* False

**BadGatewayException**

*message*

*Type:* string
Properties

Empty

GatewayTimeoutException

message

Type: string
Required: False

Input

sources

A list of the sources of the input (PULL-type).

Type: Array of type InputSource (p. 705)
Required: False

destinations

A list of the destinations of the input (PUSH-type).

Type: Array of type InputDestination (p. 704)
Required: False

type

Type: InputType (p. 706)
Required: False

tags

A collection of key-value pairs.

Type: Tags (p. 707)
Required: False

mediaConnectFlows

A list of MediaConnect Flows for this input.

Type: Array of type MediaConnectFlow (p. 706)
Required: False

attachedChannels

A list of channel IDs that that input is attached to (currently an input can only be attached to one channel).

Type: Array of type string
**Properties**

- **roleArn**
  The Amazon Resource Name (ARN) of the role this input assumes during and after creation.
  
  **Type:** string
  **Required:** False

- **inputClass**
  STANDARD - MediaLive expects two sources to be connected to this input. If the channel is also STANDARD, both sources will be ingested. If the channel is SINGLE_PIPELINE, only the first source will be ingested; the second source will always be ignored, even if the first source fails. SINGLE_PIPELINE - You can connect only one source to this input. If the ChannelClass is also SINGLE_PIPELINE, this value is valid. If the ChannelClass is STANDARD, this value is not valid because the channel requires two sources in the input.
  
  **Type:** InputClass (p. 704)
  **Required:** False

- **name**
  The user-assigned name (This is a mutable value).
  
  **Type:** string
  **Required:** False

- **securityGroups**
  A list of IDs for all the Input Security Groups attached to the input.
  
  **Type:** Array of type string
  **Required:** False

- **id**
  The generated ID of the input (unique for user account, immutable).
  
  **Type:** string
  **Required:** False

- **state**
  
  **Type:** InputState (p. 706)
  **Required:** False

- **arn**
  The Unique ARN of the input (generated, immutable).
  
  **Type:** string
  **Required:** False
InputClass

A standard input has two sources and a single pipeline input only has one.

STANDARD
SINGLE_PIPELINE

InputDestination

The settings for a PUSH type input.

port

The port number for the input.

Type: string
Required: False

ip

The system-generated static IP address of endpoint. It remains fixed for the lifetime of the input.

Type: string
Required: False

vpc

Type: InputDestinationVpc (p. 704)
Required: False

url

This represents the endpoint that the customer stream will be pushed to.

Type: string
Required: False

InputDestinationRequest

Endpoint settings for a PUSH type input.

streamName

A unique name for the location the RTMP stream is being pushed to.

Type: string
Required: False

InputDestinationVpc

The properties for a VPC type input destination.
**networkInterfaceId**

The network interface ID of the Input destination in the VPC.

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

**availabilityZone**

The availability zone of the Input destination.

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

**InputSource**

The settings for a PULL type input.

**passwordParam**

The key used to extract the password from EC2 Parameter store.

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

**url**

This represents the customer's source URL where stream is pulled from.

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

**username**

The username for the input source.

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

**InputSourceRequest**

Settings for a PULL type input.

**passwordParam**

The key used to extract the password from EC2 Parameter store.

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

**url**

This represents the customer's source URL where stream is pulled from.

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

The username for the input source.

*Type*: string  
*Required*: False

**InputState**

- CREATING
- DETACHED
- ATTACHED
- DELETING
- DELETED

**InputType**

- UDP_PUSH
- RTP_PUSH
- RTMP_PUSH
- RTMP_PULL
- URL_PULL
- MP4_FILE
- MEDIACONNECT

**InternalServiceError**

**message**

*Type*: string  
*Required*: False

**InvalidRequest**

**message**

*Type*: string  
*Required*: False

**LimitExceeded**

**message**

*Type*: string  
*Required*: False

**MediaConnectFlow**

The settings for a MediaConnect Flow.
**flowArn**
The unique ARN of the MediaConnect Flow being used as a source.

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

**MediaConnectFlowRequest**
The settings for a MediaConnect Flow.

**flowArn**
The ARN of the MediaConnect Flow that you want to use as a source.

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

**ResourceConflict**

**message**

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

**ResourceNotFound**

**message**

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

**Tags**

**key-value pairs**

- **Type:** string

**UpdateInput**

**inputSecurityGroups**

A list of security groups referenced by IDs to attach to the input.

- **Type:** Array of type string
- **Required:** False

**mediaConnectFlows**

A list of the MediaConnect Flow ARNs that you want to use as the source of the input. You can specify as few as one Flow and presently, as many as two. The only requirement is when you have more than one is that each Flow is in a separate Availability Zone as this ensures your EML input is redundant to AZ issues.
**Type:** Array of type MediaConnectFlowRequest (p. 707)  
**Required:** False

**sources**

The source URLs for a PULL-type input. Every PULL type input needs exactly two source URLs for redundancy. Only specify sources for PULL type Inputs. Leave Destinations empty.

**Type:** Array of type InputSourceRequest (p. 705)  
**Required:** False

**roleArn**

The Amazon Resource Name (ARN) of the role this input assumes during and after creation.

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

**destinations**

Destination settings for PUSH type inputs.

**Type:** Array of type InputDestinationRequest (p. 704)  
**Required:** False

**name**

Name of the input.

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

**UpdateInputResultModel**

**input**

**Type:** Input (p. 702)  
**Required:** False

**Offerings**

**URI**

/prod/offering

**HTTP Methods**

**GET**

Operation ID: ListOfferings

List offerings available for purchase.
## Query Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>resourceType</td>
<td>String</td>
<td>False</td>
<td>Filter by resource type, 'INPUT', 'OUTPUT', or 'CHANNEL'</td>
</tr>
<tr>
<td>nextPageToken</td>
<td>String</td>
<td>False</td>
<td>Filter to offerings that match the configuration of an existing channel, e.g. '2345678' (a channel ID)</td>
</tr>
<tr>
<td>channelConfiguration</td>
<td>String</td>
<td>False</td>
<td>Filter to offerings that match the configuration of an existing channel, e.g. '2345678' (a channel ID)</td>
</tr>
<tr>
<td>codec</td>
<td>String</td>
<td>False</td>
<td>Filter by codec, 'AVC', 'HEVC', 'MPEG2', or 'AUDIO'</td>
</tr>
<tr>
<td>videoQuality</td>
<td>String</td>
<td>False</td>
<td>Filter by video quality, 'STANDARD', 'ENHANCED', or 'PREMIUM'</td>
</tr>
<tr>
<td>resolution</td>
<td>String</td>
<td>False</td>
<td>Filter by resolution, 'SD', 'HD', or 'UHD'</td>
</tr>
<tr>
<td>maximumFramerate</td>
<td>String</td>
<td>False</td>
<td>Filter by framerate, 'MAX_30_FPS' or 'MAX_60_FPS'</td>
</tr>
<tr>
<td>channelClass</td>
<td>String</td>
<td>False</td>
<td>Filter by channel class, 'STANDARD' or 'SINGLE_PIPELINE'</td>
</tr>
<tr>
<td>maxResults</td>
<td>String</td>
<td>False</td>
<td>Filter by bitrate, 'MAX_10_MBPS', 'MAX_20_MBPS', or 'MAX_50_MBPS'</td>
</tr>
<tr>
<td>maximumBitrate</td>
<td>String</td>
<td>False</td>
<td>Filter by special feature, 'ADVANCED_AUDIO' or 'AUDIO_NORMALIZATION'</td>
</tr>
<tr>
<td>specialFeature</td>
<td>String</td>
<td>False</td>
<td>Filter by special feature, 'ADVANCED_AUDIO' or 'AUDIO_NORMALIZATION'</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>ListOfferingsResultModel (p. 710)</td>
<td>List of offerings</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 710)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 710)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 711)</td>
<td>Request limit exceeded on list channel calls to channel service.</td>
</tr>
</tbody>
</table>
### Schemas

#### Response Bodies

**Example ListOfferingsResultModel**

```json
{
   "nextToken": "string",
   "offerings": [
   {
      "duration": integer,
      "usagePrice": number,
      "offeringType": enum,
      "resourceSpecification": {
         "codec": enum,
         "maximumFramerate": enum,
         "resolution": enum,
         "maximumBitrate": enum,
         "specialFeature": enum,
         "channelClass": enum,
         "resourceType": enum,
         "videoQuality": enum
      },
      "durationUnits": enum,
      "offeringDescription": "string",
      "arn": "string",
      "offeringId": "string",
      "region": "string",
      "fixedPrice": number,
      "currencyCode": "string"
   }
   ]
}
```

**Example InvalidRequest**

```json
{
   "message": "string"
}
```

**Example AccessDenied**

```json
{
   "message": "string"
}
```
Example LimitExceeded

```json
{
    "message": "string"
}
```

Example InternalServiceError

```json
{
    "message": "string"
}
```

Example BadGatewayException

```json
{
    "message": "string"
}
```

Example GatewayTimeoutException

```json
{
    "message": "string"
}
```

Properties

AccessDenied

message

Type: string
Required: False

BadGatewayException

message

Type: string
Required: False

ChannelClass

A standard channel has two encoding pipelines and a single pipeline channel only has one.

STANDARD
SINGLE_PIPELINE

GatewayTimeoutException

message

Type: string
Required: False

**InternalServiceError**

message

Type: string
Required: False

**InvalidRequest**

message

Type: string
Required: False

**LimitExceeded**

message

Type: string
Required: False

**ListOfferingsResultModel**

ListOfferings response

**nextToken**

Token to retrieve the next page of results

Type: string
Required: False

**offerings**

List of offerings

Type: Array of type Offering (p. 712)
Required: False

**Offering**

Reserved resources available for purchase

**duration**

Lease duration, e.g. '12'

Type: integer
Required: False
usagePrice
Recurring usage charge for each reserved resource, e.g. '157.0'
  Type: number
  Required: False

offeringType
Offering type, e.g. 'NO_UPFRONT'
  Type: OfferingType (p. 714)
  Required: False

resourceSpecification
Resource configuration details
  Type: ReservationResourceSpecification (p. 715)
  Required: False

durationUnits
Units for duration, e.g. 'MONTHS'
  Type: OfferingDurationUnits (p. 714)
  Required: False

offeringDescription
Offering description, e.g. 'HD AVC output at 10-20 Mbps, 30 fps, and standard VQ in US West (Oregon)'
  Type: string
  Required: False

arn
Unique offering ARN, e.g. 'arn:aws:medialive:us-west-2:123456789012:offering:87654321'
  Type: string
  Required: False

offeringId
Unique offering ID, e.g. '87654321'
  Type: string
  Required: False

region
AWS region, e.g. 'us-west-2'
  Type: string
**fixedPrice**

One-time charge for each reserved resource, e.g. '0.0' for a NO_UPFRONT offering

- **Type**: number
- **Required**: False

**currencyCode**

Currency code for usagePrice and fixedPrice in ISO-4217 format, e.g. 'USD'

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

**OfferingDurationUnits**

Units for duration, e.g. 'MONTHS'

- **Value**: MONTHS

**OfferingType**

Offering type, e.g. 'NO_UPFRONT'

- **Value**: NO_UPFRONT

**ReservationCodec**

Codec, 'MPEG2', 'AVC', 'HEVC', or 'AUDIO'

- **Value**: MPEG2
- **Value**: AVC
- **Value**: HEVC
- **Value**: AUDIO

**ReservationMaximumBitrate**

Maximum bitrate in megabits per second

- **Value**: MAX_10_MBPS
- **Value**: MAX_20_MBPS
- **Value**: MAX_50_MBPS

**ReservationMaximumFramerate**

Maximum framerate in frames per second (Outputs only)

- **Value**: MAX_30_FPS
- **Value**: MAX_60_FPS
ReservationResolution

Resolution based on lines of vertical resolution; SD is less than 720 lines, HD is 720 to 1080 lines, UHD is greater than 1080 lines

- SD
- HD
- UHD

ReservationResourceSpecification

Resource configuration (codec, resolution, bitrate, ...)

**codec**

Codec, e.g. 'AVC'

*Type: ReservationCodec (p. 714)*  
*Required: False*

**maximumFramerate**

Maximum framerate, e.g. 'MAX_30_FPS' (Outputs only)

*Type: ReservationMaximumFramerate (p. 714)*  
*Required: False*

**resolution**

Resolution, e.g. 'HD'

*Type: ReservationResolution (p. 715)*  
*Required: False*

**maximumBitrate**

Maximum bitrate, e.g. 'MAX_20_MBPS'

*Type: ReservationMaximumBitrate (p. 714)*  
*Required: False*

**specialFeature**

Special feature, e.g. 'AUDIO_NORMALIZATION' (Channels only)

*Type: ReservationSpecialFeature (p. 716)*  
*Required: False*

**channelClass**

Channel class, e.g. 'STANDARD'

*Type: ChannelClass (p. 711)*
Offerings offeringId

**Required:** False

**resourceType**

Resource type, 'INPUT', 'OUTPUT', or 'CHANNEL'

**Type:** ReservationResourceType (p. 716)

**Required:** False

**videoQuality**

Video quality, e.g. 'STANDARD' (Outputs only)

**Type:** ReservationVideoQuality (p. 716)

**Required:** False

**ReservationResourceType**

Resource type, 'INPUT', 'OUTPUT', or 'CHANNEL'

- INPUT
- OUTPUT
- CHANNEL

**ReservationSpecialFeature**

Special features, 'ADVANCED_AUDIO' or 'AUDIO_NORMALIZATION'

- ADVANCED_AUDIO
- AUDIO_NORMALIZATION

**ReservationVideoQuality**

Video quality, e.g. 'STANDARD' (Outputs only)

- STANDARD
- ENHANCED
- PREMIUM

**Offerings offeringId**

**URI**

/prod/offeringsofferingId

**HTTP Methods**

**GET**

Operation ID: DescribeOffering
Get details for an offering.

**Path Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>offeringId</td>
<td>String</td>
<td>True</td>
<td>Unique offering ID, e.g. '87654321'</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>Offering (p. 717)</td>
<td>Offering details</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 718)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 718)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>404</td>
<td>ResourceNotFound (p. 718)</td>
<td>The channel you're requesting to describe does not exist.</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 718)</td>
<td>Request limit exceeded on list channel calls to channel service.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 718)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>BadGatewayException (p. 718)</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>504</td>
<td>GatewayTimeoutException (p. 718)</td>
<td>Gateway Timeout Error</td>
</tr>
</tbody>
</table>

**Schemas**

**Response Bodies**

**Example Offering**

```json
{
   "duration": integer,
   "usagePrice": number,
   "offeringType": enum,
   "resourceSpecification": {
      "codec": enum,
      "maximumFramerate": enum,
      "resolution": enum,
      "maximumBitrate": enum,
      "specialFeature": enum,
      "channelClass": enum,
      "resourceType": enum,
      "videoQuality": enum
   },
   "durationUnits": enum,
   "offeringDescription": "string",
   "arn": "string",
   "offeringId": "string",
   "region": "string",
   "fixedPrice": number
}
```
"currencyCode": "string"
}

Example InvalidRequest

{
   "message": "string"
}

Example AccessDenied

{
   "message": "string"
}

Example ResourceNotFound

{
   "message": "string"
}

Example LimitExceeded

{
   "message": "string"
}

Example InternalServiceError

{
   "message": "string"
}

Example BadGatewayException

{
   "message": "string"
}

Example GatewayTimeoutException

{
   "message": "string"
}

Properties

AccessDenied

message

Type: string
Required: False
**BadGatewayException**

`message`

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

**ChannelClass**

A standard channel has two encoding pipelines and a single pipeline channel only has one.

- STANDARD
- SINGLE_PIPELINE

**GatewayTimeoutException**

`message`

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

**InternalServiceError**

`message`

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

**InvalidRequest**

`message`

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

**LimitExceeded**

`message`

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

**Offering**

Reserved resources available for purchase

- **duration**
  - Lease duration, e.g. '12'
  - **Type**: integer
Required: False

usagePrice
Recurring usage charge for each reserved resource, e.g. '157.0'
  Type: number
  Required: False

offeringType
Offering type, e.g. 'NO_UPFRONT'
  Type: OfferingType (p. 721)
  Required: False

resourceSpecification
Resource configuration details
  Type: ReservationResourceSpecification (p. 722)
  Required: False

durationUnits
Units for duration, e.g. 'MONTHS'
  Type: OfferingDurationUnits (p. 721)
  Required: False

offeringDescription
Offering description, e.g. 'HD AVC output at 10-20 Mbps, 30 fps, and standard VQ in US West (Oregon)'
  Type: string
  Required: False

arn
Unique offering ARN, e.g. 'arn:aws:medialive:us-west-2:123456789012:offering:87654321'
  Type: string
  Required: False

offeringId
Unique offering ID, e.g. '87654321'
  Type: string
  Required: False

region
AWS region, e.g. 'us-west-2'
Properties

**Type**
string

**Required**
False

**fixedPrice**
One-time charge for each reserved resource, e.g. '0.0' for a NO_UPFRONT offering

**Type**
number

**Required**
False

**currencyCode**
Currency code for usagePrice and fixedPrice in ISO-4217 format, e.g. 'USD'

**Type**
string

**Required**
False

**OfferingDurationUnits**
Units for duration, e.g. 'MONTHS'

MONTHS

**OfferingType**
Offering type, e.g. 'NO_UPFRONT'

NO_UPFRONT

**ReservationCodec**
Codec, 'MPEG2', 'AVC', 'HEVC', or 'AUDIO'

MPEG2
AVC
HEVC
AUDIO

**ReservationMaximumBitrate**
Maximum bitrate in megabits per second

MAX_10_MBPS
MAX_20_MBPS
MAX_50_MBPS

**ReservationMaximumFramerate**
Maximum framerate in frames per second (Outputs only)

MAX_30_FPS
MAX_60_FPS
### ReservationResolution
Resolution based on lines of vertical resolution; SD is less than 720 lines, HD is 720 to 1080 lines, UHD is greater than 1080 lines

- SD
- HD
- UHD

### ReservationResourceSpecification
Resource configuration (codec, resolution, bitrate, ...)

#### codec
Codec, e.g. 'AVC'

- **Type**: ReservationCodec (p. 721)
- **Required**: False

#### maximumFramerate
Maximum framerate, e.g. 'MAX_30_FPS' (Outputs only)

- **Type**: ReservationMaximumFramerate (p. 721)
- **Required**: False

#### resolution
Resolution, e.g. 'HD'

- **Type**: ReservationResolution (p. 722)
- **Required**: False

#### maximumBitrate
Maximum bitrate, e.g. 'MAX_20_MBPS'

- **Type**: ReservationMaximumBitrate (p. 721)
- **Required**: False

#### specialFeature
Special feature, e.g. 'AUDIO_NORMALIZATION' (Channels only)

- **Type**: ReservationSpecialFeature (p. 723)
- **Required**: False

#### channelClass
Channel class, e.g. 'STANDARD'

- **Type**: ChannelClass (p. 719)
- **Required**: False
resourceType

Resource type, 'INPUT', 'OUTPUT', or 'CHANNEL'

Type: ReservationResourceType (p. 723)
Required: False

videoQuality

Video quality, e.g. 'STANDARD' (Outputs only)

Type: ReservationVideoQuality (p. 723)
Required: False

ReservationResourceType

Resource type, 'INPUT', 'OUTPUT', or 'CHANNEL'

INPUT
OUTPUT
CHANNEL

ReservationSpecialFeature

Special features, 'ADVANCED_AUDIO' or 'AUDIO_NORMALIZATION'

ADVANCED_AUDIO
AUDIO_NORMALIZATION

ReservationVideoQuality

Video quality, e.g. 'STANDARD' (Outputs only)

STANDARD
ENHANCED
PREMIUM

ResourceNotFound

message

Type: string
Required: False

Offerings offeringId Purchase

URI

/prod/offerings/offeringId/purchase
HTTP Methods

POST

Operation ID: PurchaseOffering

Purchase an offering and create a reservation.

Path Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>offeringId</td>
<td>String</td>
<td>True</td>
<td>Unique offering ID, e.g. '87654321'</td>
</tr>
</tbody>
</table>

Responses

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>PurchaseOfferingResultModel</td>
<td>Creation of channel is started.</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 725)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 725)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>404</td>
<td>ResourceNotFound (p. 725)</td>
<td>The channel you're requesting to describe does not exist.</td>
</tr>
<tr>
<td>409</td>
<td>ResourceConflict (p. 726)</td>
<td>The channel is unable to create due to an issue with channel resources.</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 726)</td>
<td>Request limit exceeded on list channel calls to channel service.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 721)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>BadGatewayException (p. 724)</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>504</td>
<td>GatewayTimeoutException (p. 725)</td>
<td>Gateway Timeout Error</td>
</tr>
</tbody>
</table>

Schemas

Request Bodies

Example POST

```json
{
  "requestId": "string",
  "count": integer,
  "name": "string",
  "start": "string",
  "tags": {
```
Response Bodies

Example PurchaseOfferingResultModel

```json
{
    "reservation": {
        "offeringType": enum,
        "count": integer,
        "start": "string",
        "resourceSpecification": {
            "codec": enum,
            "maximumFramerate": enum,
            "resolution": enum,
            "maximumBitrate": enum,
            "specialFeature": enum,
            "channelClass": enum,
            "resourceType": enum,
            "videoQuality": enum
        },
        "durationUnits": enum,
        "offeringId": "string",
        "fixedPrice": number,
        "tags": {
        },
        "duration": integer,
        "usagePrice": number,
        "reservationId": "string",
        "name": "string",
        "end": "string",
        "state": enum,
        "offeringDescription": "string",
        "arn": "string",
        "region": "string",
        "currencyCode": "string"
    }
}
```

Example InvalidRequest

```json
{
    "message": "string"
}
```

Example AccessDenied

```json
{
    "message": "string"
}
```

Example ResourceNotFound

```json
{
    "message": "string"
}
```
Example ResourceConflict

{  "message": "string"
}

Example LimitExceeded

{  "message": "string"
}

Example InternalServiceError

{  "message": "string"
}

Example BadGatewayException

{  "message": "string"
}

Example GatewayTimeoutException

{  "message": "string"
}

Properties

AccessDenied

message

  Type: string
  Required: False

BadGatewayException

message

  Type: string
  Required: False

ChannelClass

A standard channel has two encoding pipelines and a single pipeline channel only has one.

  STANDARD
  SINGLE_PIPELINE
GatewayTimeoutException

message
  Type: string
  Required: False

InternalServiceError

message
  Type: string
  Required: False

InvalidRequest

message
  Type: string
  Required: False

LimitExceeded

message
  Type: string
  Required: False

OfferingDurationUnits

Units for duration, e.g. 'MONTHS'
  MONTHS

OfferingType

Offering type, e.g. 'NO_UPFRONT'
  NO_UPFRONT

PurchaseOffering

PurchaseOffering request

requestId

Unique request ID to be specified. This is needed to prevent retries from creating multiple resources.
  Type: string
  Required: False
**count**

Number of resources

- **Type:** integer
- **Required:** True
- **Minimum:** 1

**name**

Name for the new reservation

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

**start**

Requested reservation start time (UTC) in ISO-8601 format. The specified time must be between the first day of the current month and one year from now. If no value is given, the default is now.

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

**tags**

A collection of key-value pairs

- **Type:** Tags (p. 733)
- **Required:** False

**PurchaseOfferingResultModel**

PurchaseOffering response

**reservation**

- **Type:** Reservation (p. 728)
- **Required:** False

**Reservation**

Reserved resources available to use

**offeringType**

Offering type, e.g. 'NO_UPFRONT'

- **Type:** OfferingType (p. 727)
- **Required:** False

**count**

Number of reserved resources

- **Type:** integer
Required: False

start
Reservation UTC start date and time in ISO-8601 format, e.g. '2018-03-01T00:00:00'

Type: string
Required: False

resourceSpecification
Resource configuration details

Type: ReservationResourceSpecification (p. 731)
Required: False

durationUnits
Units for duration, e.g. 'MONTHS'

Type: OfferingDurationUnits (p. 727)
Required: False

offeringId
Unique offering ID, e.g. '87654321'

Type: string
Required: False

fixedPrice
One-time charge for each reserved resource, e.g. '0.0' for a NO_UPFRONT offering

Type: number
Required: False

tags
A collection of key-value pairs

Type: Tags (p. 733)
Required: False

duration
Lease duration, e.g. '12'

Type: integer
Required: False

usagePrice
Recurring usage charge for each reserved resource, e.g. '157.0'
properties

reservationId
Unique reservation ID, e.g. '1234567'

name
User specified reservation name

date
Reservation UTC end date and time in ISO-8601 format, e.g. '2019-03-01T00:00:00'

state
Current state of reservation, e.g. 'ACTIVE'

offeringDescription
Offering description, e.g. 'HD AVC output at 10-20 Mbps, 30 fps, and standard VQ in US West (Oregon)'

arn
Unique reservation ARN, e.g. 'arn:aws:medialive:us-west-2::123456789012:reservation:1234567'

region
AWS region, e.g. 'us-west-2'

currencyCode
Currency code for usagePrice and fixedPrice in ISO-4217 format, e.g. 'USD'
Properties

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

**ReservationCodec**

Codec, 'MPEG2', 'AVC', 'HEVC', or 'AUDIO'

- MPEG2
- AVC
- HEVC
- AUDIO

**ReservationMaximumBitrate**

Maximum bitrate in megabits per second

- MAX_10_MBPS
- MAX_20_MBPS
- MAX_50_MBPS

**ReservationMaximumFramerate**

Maximum framerate in frames per second (Outputs only)

- MAX_30_FPS
- MAX_60_FPS

**ReservationResolution**

Resolution based on lines of vertical resolution; SD is less than 720 lines, HD is 720 to 1080 lines, UHD is greater than 1080 lines

- SD
- HD
- UHD

**ReservationResourceSpecification**

Resource configuration (codec, resolution, bitrate, ...)

**codec**

Codec, e.g. 'AVC'

**Type**: ReservationCodec (p. 731)  
**Required**: False

**maximumFramerate**

Maximum framerate, e.g. 'MAX_30_FPS' (Outputs only)
Properties

Type: ReservationMaximumFramerate (p. 731)
Required: False

resolution
Resolution, e.g. 'HD'
Type: ReservationResolution (p. 731)
Required: False

maximumBitrate
Maximum bitrate, e.g. 'MAX_20_MBPS'
Type: ReservationMaximumBitrate (p. 731)
Required: False

specialFeature
Special feature, e.g. 'AUDIO_NORMALIZATION' (Channels only)
Type: ReservationSpecialFeature (p. 733)
Required: False

channelClass
Channel class, e.g. 'STANDARD'
Type: ChannelClass (p. 726)
Required: False

resourceType
Resource type, 'INPUT', 'OUTPUT', or 'CHANNEL'
Type: ReservationResourceType (p. 732)
Required: False

videoQuality
Video quality, e.g. 'STANDARD' (Outputs only)
Type: ReservationVideoQuality (p. 733)
Required: False

ReservationResourceType
Resource type, 'INPUT', 'OUTPUT', or 'CHANNEL'

INPUT
OUTPUT
CHANNEL
**ReservationSpecialFeature**

Special features, 'ADVANCED_AUDIO' or 'AUDIO_NORMALIZATION'

- ADVANCED_AUDIO
- AUDIO_NORMALIZATION

**ReservationState**

Current reservation state

- ACTIVE
- EXPIRED
- CANCELED
- DELETED

**ReservationVideoQuality**

Video quality, e.g. 'STANDARD' (Outputs only)

- STANDARD
- ENHANCED
- PREMIUM

**ResourceConflict**

`message`

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

**ResourceNotFound**

`message`

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

**Tags**

- **key-value pairs**
- **Type:** string

**Reservations**

**URI**

/prod/reservations
HTTP Methods

GET

Operation ID: ListReservations

List purchased reservations.

Query Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>resourceType</td>
<td>String</td>
<td>False</td>
<td>Filter by resource type, 'INPUT', 'OUTPUT', or 'CHANNEL'</td>
</tr>
<tr>
<td>nextToken</td>
<td>String</td>
<td>False</td>
<td></td>
</tr>
<tr>
<td>codec</td>
<td>String</td>
<td>False</td>
<td>Filter by codec, 'AVC', 'HEVC', 'MPEG2', or 'AUDIO'</td>
</tr>
<tr>
<td>videoQuality</td>
<td>String</td>
<td>False</td>
<td>Filter by video quality, 'STANDARD', 'ENHANCED', or 'PREMIUM'</td>
</tr>
<tr>
<td>resolution</td>
<td>String</td>
<td>False</td>
<td>Filter by resolution, 'SD', 'HD', or 'UHD'</td>
</tr>
<tr>
<td>maximumFramerate</td>
<td>String</td>
<td>False</td>
<td>Filter by framerate, 'MAX_30_FPS' or 'MAX_60_FPS'</td>
</tr>
<tr>
<td>channelClass</td>
<td>String</td>
<td>False</td>
<td>Filter by channel class, 'STANDARD' or 'SINGLE_PIPELINE'</td>
</tr>
<tr>
<td>maxResults</td>
<td>String</td>
<td>False</td>
<td></td>
</tr>
<tr>
<td>maximumBitrate</td>
<td>String</td>
<td>False</td>
<td>Filter by bitrate, 'MAX_10_MBPS', 'MAX_20_MBPS', or 'MAX_50_MBPS'</td>
</tr>
<tr>
<td>specialFeature</td>
<td>String</td>
<td>False</td>
<td>Filter by special feature, 'ADVANCED_AUDIO' or 'AUDIO_NORMALIZATION'</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>ListReservationsResultModel (p. 735)</td>
<td>List of reservations</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 735)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 736)</td>
<td>You do not have permission to list channels.</td>
</tr>
</tbody>
</table>
### Status Code

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>429</td>
<td>LimitExceeded (p. 736)</td>
<td>Request limit exceeded on list channel calls to channel service.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 736)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>BadGatewayException (p. 736)</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>504</td>
<td>GatewayTimeoutException (p. 736)</td>
<td>Gateway Timeout Error</td>
</tr>
</tbody>
</table>

### Schemas

#### Response Bodies

**Example ListReservationsResultModel**

```json
{
    "reservations": [
        {
            "offeringType": enum,
            "count": integer,
            "start": "string",
            "resourceSpecification": {
                "codec": enum,
                "maximumFramerate": enum,
                "resolution": enum,
                "maximumBitrate": enum,
                "specialFeature": enum,
                "channelClass": enum,
                "resourceType": enum,
                "videoQuality": enum
            },
            "durationUnits": enum,
            "offeringId": "string",
            "fixedPrice": number,
            "tags": {
            },
            "duration": integer,
            "usagePrice": number,
            "reservationId": "string",
            "name": "string",
            "end": "string",
            "state": enum,
            "offeringDescription": "string",
            "arn": "string",
            "region": "string",
            "currencyCode": "string"
        }
    ],
    "nextToken": "string"
}
```

**Example InvalidRequest**

```json
{
    "message": "string"
}
```
Example AccessDenied

```json
{
  "message": "string"
}
```

Example LimitExceeded

```json
{
  "message": "string"
}
```

Example InternalServiceError

```json
{
  "message": "string"
}
```

Example BadGatewayException

```json
{
  "message": "string"
}
```

Example GatewayTimeoutException

```json
{
  "message": "string"
}
```

Properties

AccessDenied

**message**

Type: string  
Required: False

BadGatewayException

**message**

Type: string  
Required: False

ChannelClass

A standard channel has two encoding pipelines and a single pipeline channel only has one.

STANDARD
SINGLE_PIPELINE

**GatewayTimeoutException**

*message*

  *Type:* string  
  *Required:* False

**InternalServiceError**

*message*

  *Type:* string  
  *Required:* False

**InvalidRequest**

*message*

  *Type:* string  
  *Required:* False

**LimitExceeded**

*message*

  *Type:* string  
  *Required:* False

**ListReservationsResultModel**

ListReservations response

*reservations*

List of reservations

  *Type:* Array of type Reservation (p. 738)  
  *Required:* False

*nextToken*

Token to retrieve the next page of results

  *Type:* string  
  *Required:* False

**OfferingDurationUnits**

Units for duration, e.g. 'MONTHS'
MONTHS

**OfferingType**

Offering type, e.g. 'NO_UPFRONT'

NO_UPFRONT

**Reservation**

Reserved resources available to use

**offeringType**

Offering type, e.g. 'NO_UPFRONT'

  *Type: OfferingType (p. 738)*
  *Required: False*

**count**

Number of reserved resources

  *Type: integer*
  *Required: False*

**start**

Reservation UTC start date and time in ISO-8601 format, e.g. '2018-03-01T00:00:00'

  *Type: string*
  *Required: False*

**resourceSpecification**

Resource configuration details

  *Type: ReservationResourceSpecification (p. 741)*
  *Required: False*

**durationUnits**

Units for duration, e.g. 'MONTHS'

  *Type: OfferingDurationUnits (p. 737)*
  *Required: False*

**offeringId**

Unique offering ID, e.g. '87654321'

  *Type: string*
  *Required: False*
fixedPrice

One-time charge for each reserved resource, e.g. '0.0' for a NO_UPFRONT offering

Type: number
Required: False

tags

A collection of key-value pairs

Type: Tags (p. 742)
Required: False

duration

Lease duration, e.g. '12'

Type: integer
Required: False

usagePrice

Recurring usage charge for each reserved resource, e.g. '157.0'

Type: number
Required: False

reservationId

Unique reservation ID, e.g. '1234567'

Type: string
Required: False

name

User specified reservation name

Type: string
Required: False

end

Reservation UTC end date and time in ISO-8601 format, e.g. '2019-03-01T00:00:00'

Type: string
Required: False

state

Current state of reservation, e.g. 'ACTIVE'

Type: ReservationState (p. 742)
Properties

offeringDescription

Offering description, e.g. 'HD AVC output at 10-20 Mbps, 30 fps, and standard VQ in US West (Oregon)'

Type: string
Required: False

arn

Unique reservation ARN, e.g. 'arn:aws:medialive:us-west-2:123456789012:reservation:1234567'

Type: string
Required: False

region

AWS region, e.g. 'us-west-2'

Type: string
Required: False

currencyCode

Currency code for usagePrice and fixedPrice in ISO-4217 format, e.g. 'USD'

Type: string
Required: False

ReservationCodec

Codec, 'MPEG2', 'AVC', 'HEVC', or 'AUDIO'

MPEG2
AVC
HEVC
AUDIO

ReservationMaximumBitrate

Maximum bitrate in megabits per second

MAX_10_MBPS
MAX_20_MBPS
MAX_50_MBPS

ReservationMaximumFramerate

Maximum framerate in frames per second (Outputs only)

MAX_30_FPS
MAX_60_FPS
**ReservationResolution**

Resolution based on lines of vertical resolution; SD is less than 720 lines, HD is 720 to 1080 lines, UHD is greater than 1080 lines

- SD
- HD
- UHD

**ReservationResourceSpecification**

Resource configuration (codec, resolution, bitrate, ...)

**codec**

Codec, e.g. 'AVC'

_Type: ReservationCodec (p. 740)_

_Required: False_

**maximumFramerate**

Maximum framerate, e.g. 'MAX_30_FPS' (Outputs only)

_Type: ReservationMaximumFramerate (p. 740)_

_Required: False_

**resolution**

Resolution, e.g. 'HD'

_Type: ReservationResolution (p. 741)_

_Required: False_

**maximumBitrate**

Maximum bitrate, e.g. 'MAX_20_MBPS'

_Type: ReservationMaximumBitrate (p. 740)_

_Required: False_

**specialFeature**

Special feature, e.g. 'AUDIO_NORMALIZATION' (Channels only)

_Type: ReservationSpecialFeature (p. 742)_

_Required: False_

**channelClass**

Channel class, e.g. 'STANDARD'

_Type: ChannelClass (p. 736)_

_Required: False_
**resourceType**

Resource type, 'INPUT', 'OUTPUT', or 'CHANNEL'

*Type: ReservationResourceType (p. 742)*

*Required: False*

**videoQuality**

Video quality, e.g. 'STANDARD' (Outputs only)

*Type: ReservationVideoQuality (p. 742)*

*Required: False*

**ReservationResourceType**

Resource type, 'INPUT', 'OUTPUT', or 'CHANNEL'

  - INPUT
  - OUTPUT
  - CHANNEL

**ReservationSpecialFeature**

Special features, 'ADVANCED_AUDIO' or 'AUDIO_NORMALIZATION'

  - ADVANCED_AUDIO
  - AUDIO_NORMALIZATION

**ReservationState**

Current reservation state

  - ACTIVE
  - EXPIRED
  - CANCELED
  - DELETED

**ReservationVideoQuality**

Video quality, e.g. 'STANDARD' (Outputs only)

  - STANDARD
  - ENHANCED
  - PREMIUM

**Tags**

key-value pairs

*Type: string*
Reservations reservationId

URI

/prod/reservations/reservationId

HTTP Methods

GET

Operation ID: DescribeReservation

Get details for a reservation.

Path Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reservationId</td>
<td>String</td>
<td>True</td>
<td>Unique reservation ID, e.g. '1234567'</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>Reservation (p. 745)</td>
<td>Reservation details</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 746)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 746)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>404</td>
<td>ResourceNotFound (p. 746)</td>
<td>The channel you're requesting to describe does not exist.</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 747)</td>
<td>Request limit exceeded on list channel calls to channel service.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 747)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>BadGatewayException (p. 747)</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>504</td>
<td>GatewayTimeoutException (p. 747)</td>
<td>Gateway Timeout Error</td>
</tr>
</tbody>
</table>

PUT

Operation ID: UpdateReservation

Update reservation.
Path Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reservationId</td>
<td>String</td>
<td>True</td>
<td>Unique reservation ID, e.g. '1234567'</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>UpdateReservationResultModel</td>
<td>Updated reservation</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 746)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 746)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>404</td>
<td>ResourceNotFound (p. 746)</td>
<td>The channel you're requesting to describe does not exist.</td>
</tr>
<tr>
<td>409</td>
<td>ResourceConflict (p. 747)</td>
<td>The channel is unable to create due to an issue with channel resources.</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 747)</td>
<td>Request limit exceeded on list channel calls to channel service.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 747)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>BadGatewayException (p. 747)</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>504</td>
<td>GatewayTimeoutException (p. 747)</td>
<td>Gateway Timeout Error</td>
</tr>
</tbody>
</table>

DELETE

Operation ID: DeleteReservation
Delete an expired reservation.

Path Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reservationId</td>
<td>String</td>
<td>True</td>
<td>Unique reservation ID, e.g. '1234567'</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>Reservation (p. 745)</td>
<td>Deleted reservation</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 746)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 746)</td>
<td>You do not have permission to list channels.</td>
</tr>
</tbody>
</table>
### Status Code

<table>
<thead>
<tr>
<th>Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>404</td>
<td>ResourceNotFound (p. 746)</td>
<td>The channel you're requesting to describe does not exist.</td>
</tr>
<tr>
<td>409</td>
<td>ResourceConflict (p. 747)</td>
<td>The channel is unable to create due to an issue with channel resources.</td>
</tr>
<tr>
<td>429</td>
<td>LimitExceeded (p. 747)</td>
<td>Request limit exceeded on list channel calls to channel service.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 747)</td>
<td>Unexpected internal service error.</td>
</tr>
<tr>
<td>502</td>
<td>BadGatewayException (p. 747)</td>
<td>Bad Gateway Error</td>
</tr>
<tr>
<td>504</td>
<td>GatewayTimeoutException (p. 747)</td>
<td>Gateway Timeout Error</td>
</tr>
</tbody>
</table>

### Schemas

**Request Bodies**

Example PUT

```json
{
    "name": "string"
}
```

**Response Bodies**

Example Reservation

```json
{
    "offeringType": enum,
    "count": integer,
    "start": "string",
    "resourceSpecification": {
        "codec": enum,
        "maximumFramerate": enum,
        "resolution": enum,
        "maximumBitrate": enum,
        "specialFeature": enum,
        "channelClass": enum,
        "resourceType": enum,
        "videoQuality": enum
    },
    "durationUnits": enum,
    "offeringId": "string",
    "fixedPrice": number,
    "tags": {
    },
    "duration": integer,
    "usagePrice": number,
    "reservationId": "string",
    "name": "string",
    "end": "string",
    "state": enum
}
```
"offeringDescription": "string",
"arn": "string",
"region": "string",
"currencyCode": "string"
}

Example UpdateReservationResultModel

{
  "reservation": {
    "offeringType": enum,
    "count": integer,
    "start": "string",
    "resourceSpecification": {
      "codec": enum,
      "maximumFramerate": enum,
      "resolution": enum,
      "maximumBitrate": enum,
      "specialFeature": enum,
      "channelClass": enum,
      "resourceType": enum,
      "videoQuality": enum
    },
    "durationUnits": enum,
    "offeringId": "string",
    "fixedPrice": number,
    "tags": {
    },
    "duration": integer,
    "usagePrice": number,
    "reservationId": "string",
    "name": "string",
    "end": "string",
    "state": enum,
    "offeringDescription": "string",
    "arn": "string",
    "region": "string",
    "currencyCode": "string"
  }
}

Example InvalidRequest

{
  "message": "string"
}

Example AccessDenied

{
  "message": "string"
}

Example ResourceNotFound

{
  "message": "string"
}
Example ResourceConflict

```json
{
  "message": "string"
}
```

Example LimitExceeded

```json
{
  "message": "string"
}
```

Example InternalServiceError

```json
{
  "message": "string"
}
```

Example BadGatewayException

```json
{
  "message": "string"
}
```

Example GatewayTimeoutException

```json
{
  "message": "string"
}
```

Properties

AccessDenied

**message**

*Type: string*
*Required: False*

BadGatewayException

**message**

*Type: string*
*Required: False*

ChannelClass

A standard channel has two encoding pipelines and a single pipeline channel only has one.

STANDARD
SINGLE_PIPELINE
GatewayTimeoutException

message

  Type: string
  Required: False

InternalServiceError

message

  Type: string
  Required: False

InvalidRequest

message

  Type: string
  Required: False

LimitExceeded

message

  Type: string
  Required: False

OfferingDurationUnits

Units for duration, e.g. 'MONTHS'

  MONTHS

OfferingType

Offering type, e.g. 'NO_UPFRONT'

  NO_UPFRONT

Reservation

Reserved resources available to use

offeringType

Offering type, e.g. 'NO_UPFRONT'

  Type: OfferingType (p. 748)
  Required: False
count
Number of reserved resources

Type: integer
Required: False

start
Reservation UTC start date and time in ISO-8601 format, e.g. '2018-03-01T00:00:00'

Type: string
Required: False

resourceSpecification
Resource configuration details

Type: ReservationResourceSpecification (p. 751)
Required: False

durationUnits
Units for duration, e.g. 'MONTHS'

Type: OfferingDurationUnits (p. 748)
Required: False

offeringId
Unique offering ID, e.g. '87654321'

Type: string
Required: False

fixedPrice
One-time charge for each reserved resource, e.g. '0.0' for a NO_UPFRONT offering

Type: number
Required: False

tags
A collection of key-value pairs

Type: Tags (p. 753)
Required: False

duration
Lease duration, e.g. '12'

Type: integer
Required: False
Properties

**usagePrice**
Recurring usage charge for each reserved resource, e.g. '157.0'

  * **Type:** number
  * **Required:** False

**reservationId**
Unique reservation ID, e.g. '1234567'

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

**name**
User specified reservation name

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

**end**
Reservation UTC end date and time in ISO-8601 format, e.g. '2019-03-01T00:00:00'

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

**state**
Current state of reservation, e.g. 'ACTIVE'

  * **Type:** ReservationState (p. 753)
  * **Required:** False

**offeringDescription**
Offering description, e.g. 'HD AVC output at 10-20 Mbps, 30 fps, and standard VQ in US West (Oregon)'

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

**arn**
Unique reservation ARN, e.g. 'arn:aws:medialive:us-west-2:123456789012:reservation:1234567'

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

**region**
AWS region, e.g. 'us-west-2'

  * **Type:** string
Properties

**currencyCode**
Currency code for usagePrice and fixedPrice in ISO-4217 format, e.g. 'USD'

*Type:* string
*Required:* False

**ReservationCodec**
Codec, 'MPEG2', 'AVC', 'HEVC', or 'AUDIO'

- MPEG2
- AVC
- HEVC
- AUDIO

**ReservationMaximumBitrate**
Maximum bitrate in megabits per second

- MAX_10_MBPS
- MAX_20_MBPS
- MAX_50_MBPS

**ReservationMaximumFramerate**
Maximum framerate in frames per second (Outputs only)

- MAX_30_FPS
- MAX_60_FPS

**ReservationResolution**
Resolution based on lines of vertical resolution; SD is less than 720 lines, HD is 720 to 1080 lines, UHD is greater than 1080 lines

- SD
- HD
- UHD

**ReservationResourceSpecification**
Resource configuration (codec, resolution, bitrate, ...)

**codec**
Codec, e.g. 'AVC'

*Type:* ReservationCodec (p. 751)
*Required:* False
**maximumFramerate**

Maximum framerate, e.g. 'MAX_30_FPS' (Outputs only)

- **Type:** ReservationMaximumFramerate (p. 751)
- **Required:** False

**resolution**

Resolution, e.g. 'HD'

- **Type:** ReservationResolution (p. 751)
- **Required:** False

**maximumBitrate**

Maximum bitrate, e.g. 'MAX_20_MBPS'

- **Type:** ReservationMaximumBitrate (p. 751)
- **Required:** False

**specialFeature**

Special feature, e.g. 'AUDIO_NORMALIZATION' (Channels only)

- **Type:** ReservationSpecialFeature (p. 753)
- **Required:** False

**channelClass**

Channel class, e.g. 'STANDARD'

- **Type:** ChannelClass (p. 747)
- **Required:** False

**resourceType**

Resource type, 'INPUT', 'OUTPUT', or 'CHANNEL'

- **Type:** ReservationResourceType (p. 752)
- **Required:** False

**videoQuality**

Video quality, e.g. 'STANDARD' (Outputs only)

- **Type:** ReservationVideoQuality (p. 753)
- **Required:** False

**ReservationResourceType**

Resource type, 'INPUT', 'OUTPUT', or 'CHANNEL'

- INPUT
OUTPUT
CHANNEL

**ReservationSpecialFeature**

Special features, 'ADVANCED_AUDIO' or 'AUDIO_NORMALIZATION'

- ADVANCED_AUDIO
- AUDIO_NORMALIZATION

**ReservationState**

Current reservation state

- ACTIVE
- EXPIRED
- CANCELED
-_DELETED

**ReservationVideoQuality**

Video quality, e.g. 'STANDARD' (Outputs only)

- STANDARD
- ENHANCED
- PREMIUM

**ResourceConflict**

*message*

Type: string  
Required: False

**ResourceNotFound**

*message*

Type: string  
Required: False

**Tags**

*key-value pairs*

Type: string

**UpdateReservation**

UpdateReservation request
name
Name of the reservation
Type: string
Required: False

UpdateReservationResultModel
UpdateReservation response

reservation
Type: Reservation (p. 748)
Required: False

Tags resource-arn

URI
/prod/tags/resource-arn

HTTP Methods
GET

Operation ID: ListTagsForResource
Produces list of tags that have been created for a resource

Path Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>resource-arn</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>TagsModel (p. 756)</td>
<td>An array of tags</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 756)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 756)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>404</td>
<td>ResourceNotFoundException (p. 756)</td>
<td>The channel you’re requesting to describe does not exist.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 756)</td>
<td>Unexpected internal service error.</td>
</tr>
</tbody>
</table>
POST

Operation ID: CreateTags

Create tags for a resource

Path Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>resource-arn</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>204</td>
<td>None</td>
<td>Successful response.</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 756)</td>
<td>This request was invalid.</td>
</tr>
<tr>
<td>403</td>
<td>AccessDenied (p. 756)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>404</td>
<td>ResourceNotFound (p. 756)</td>
<td>The channel you’re requesting to describe does not exist.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 756)</td>
<td>Unexpected internal service error.</td>
</tr>
</tbody>
</table>

DELETE

Operation ID: DeleteTags

Removes tags for a resource

Path Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>resource-arn</td>
<td>String</td>
<td>True</td>
<td></td>
</tr>
</tbody>
</table>

Query Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tagKeys</td>
<td>String</td>
<td>True</td>
<td>An array of tag keys to delete</td>
</tr>
</tbody>
</table>

Responses

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Response Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>204</td>
<td>None</td>
<td>Successful response.</td>
</tr>
<tr>
<td>400</td>
<td>InvalidRequest (p. 756)</td>
<td>This request was invalid.</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>403</td>
<td>AccessDenied (p. 756)</td>
<td>You do not have permission to list channels.</td>
</tr>
<tr>
<td>404</td>
<td>ResourceNotFound (p. 756)</td>
<td>The channel you're requesting to describe does not exist.</td>
</tr>
<tr>
<td>500</td>
<td>InternalServiceError (p. 756)</td>
<td>Unexpected internal service error.</td>
</tr>
</tbody>
</table>

## Schemas

### Request Bodies

#### Example POST

```json
{
  "tags": {
  }
}
```

### Response Bodies

#### Example TagsModel

```json
{
  "tags": {
  }
}
```

#### Example InvalidRequest

```json
{
  "message": "string"
}
```

#### Example AccessDenied

```json
{
  "message": "string"
}
```

#### Example ResourceNotFound

```json
{
  "message": "string"
}
```

#### Example InternalServiceError

```json
{
}
```
"message": "string"
}

Properties

AccessDenied

message

Type: string
Required: False

InternalServerError

message

Type: string
Required: False

InvalidRequest

message

Type: string
Required: False

ResourceNotFoundException

message

Type: string
Required: False

Tags

key-value pairs

Type: string

TagsModel

tags

Type: Tags (p. 757)
Required: False
## Document History

The following table describes important changes to this documentation.

- **API version: latest**

<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
</table>
| UpdateChannelClass operation in the channel, and more | Documentation for the UpdateChannelClass operation in the channel  
Documentation for the MediaPackage output group type  
Documentation for pausing and unpausing a channel using the channel schedule  
Documentation for the RTP push input and RTMP push input connected to an upstream system that is in your Amazon VPC  
Documentation for tagging using MediaLive  
Documentation for the frame capture output group | May 2, 2019      |
| Integration with AWS Elemental MediaConnect, and more | Documentation for the MediaConnect input type in the Channel resource  
Documentation for Input switching using the channel schedule  
Documentation for Schedule feature in the Channel resource  
Documentation for Reservations resource  
Documentation for the RTMP output type in the Channel resource | December 7, 2018 |
| New AWS Elemental MediaLive service release | Initial documentation for the MediaLive service.                                                                                                                                                           | November 27, 2017|
AWS Glossary

For the latest AWS terminology, see the AWS Glossary in the AWS General Reference.